

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

PROYECTO FIN DE CARRERA

**GUÍA DE RECURSOS SOBRE DAÑO CEREBRAL
ADQUIRIDO**

Autor: Jaime Aranaz Tudela

Tutora: Ana María Iglesias Maqueda

Julio 2006

*A Elena, por estar siempre ahí
para darme su apoyo y cariño.
Espero saber corresponderte.*

AGRADECIMIENTOS

Este documento no sólo representa un duro año de trabajo y esfuerzo en la realización del Proyecto Fin de Carrera, sino que también significa el término de toda una carrera universitaria y el fin de una etapa en mi vida. Es por ello que me gustaría desde estas líneas agradecer y recordar a todos aquellos que, en mayor o menor medida, me han ayudado a llegar hasta aquí.

En primer lugar y por supuesto, a mis padres, no sólo por proporcionarme unos estudios sino también por la educación que han sabido inculcarme desde que era tan sólo un niño y que me ha formado como persona. Igualmente, a mis hermanos Enrique, Carolina, Javier y Vanessa, porque sin ellos tampoco sería la persona que he llegado a ser.

A mi tutora, Ana María Iglesias, por su impagable ayuda, valiosa orientación y excelente trato con el que me ha obsequiado durante estos meses. Estoy convencido de que será tan buena madre como la buena profesora y amiga que es.

A todos mis compañeros de CESyA, por el valor humano y profesional que representan. Gracias por hacer que mi trabajo allí se convirtiera en un placer.

Durante estos años de universidad he conocido a mucha gente y de todo tipo. Afortunadamente para mí, me he encontrado con la mejor. A Antonio, Carlos, Gustavo, Martín, Nacho L. y Santiago, a vosotros os perdí pronto la pista, pero fuisteis mis primeros amigos en la universidad. Alberto M., Alex, Álvaro, Elena G., Fernando, Irene, Jessica, Juanma O., Juanra, Mari Ángeles, Mayte, Nacho H., Rodrigo, y otros a los que seguro olvido, gracias por vuestra amistad y los buenos momentos juntos.

Y entre todos ellos, gracias Elena, por haber venido a estudiar a esta universidad y haberme dado con ello la posibilidad de conocerte.

ÍNDICE DE CONTENIDOS

1	INTRODUCCIÓN	10
1.1	Motivación	10
1.2	Objetivos.....	11
1.3	Estructura del documento	13
2	GESTIÓN DEL PROYECTO	15
2.1	Gestión del software	15
2.1.1	Metodología.....	15
2.1.2	Ciclo de vida.....	16
2.1.3	Recursos software y hardware.....	17
2.1.4	Recursos humanos	18
2.2	Organización del trabajo.....	20
2.3	Planificación inicial.....	21
2.3.1	Diagrama PERT de actividades.....	21
2.3.2	Diagrama GANTT	23
3	ESTADO DEL ARTE	27
3.1	Un caso real: desarrollo de una Web accesible.....	27
3.2	Federación Española de Daño Cerebral (FEDACE)	30
3.3	Centro Español de Subtitulado y Audiodescripción (CESyA)	32
3.4	Accesibilidad Web	36
3.4.1	Pautas de Accesibilidad al Contenido en la Web	36
3.4.2	Herramientas de validación.....	39
3.4.3	Recursos Web sobre accesibilidad.....	40
3.5	Lenguajes de acceso a bases de datos.....	42
3.5.1	ASP	44
3.5.2	JSP	44
3.5.3	PHP	46
3.5.4	Perl.....	47
3.5.5	Elección de un lenguaje de acceso a bases de datos	48
3.6	Sistemas Gestores de Bases de Datos (SGBD)	49
3.6.1	Microsoft SQL Server.....	49
3.6.2	Oracle	50
3.6.3	Firebird.....	51
3.6.4	MySQL.....	52
3.6.5	PostgreSQL	53
3.6.6	SQLite	54
3.6.7	Elección de un SGBD	55
3.7	Servidores Web	55
3.7.1	Apache	56
3.7.2	IIS	56
3.7.3	Cherokee	57
3.7.4	Elección de un servidor Web.....	58

4	TOMA DE REQUISITOS Y ANÁLISIS	59
4.1	Contacto con FEDACE.....	59
4.2	Preparativos a la reunión.....	59
4.3	Exposición del problema	60
4.4	Análisis de los requisitos de usuario	62
4.4.1	Análisis de la base de datos.....	63
4.4.1.1	Requisitos específicos de usuario sobre la base de datos.....	63
4.4.1.2	Análisis de los datos.....	63
4.4.1.3	Análisis del SGBD.....	68
4.4.2	Análisis de la aplicación Web de gestión de la base de datos.....	68
4.4.2.1	Requisitos de usuario sobre la aplicación Web	68
4.4.2.2	Análisis de los tipos de usuario	69
4.4.2.3	Análisis de las funcionalidades.....	70
4.4.2.4	Análisis de la accesibilidad Web	70
4.4.2.5	Publicación en Internet.....	71
5	DISEÑO DE LA BASE DE DATOS	73
5.1	Modelo conceptual: obtención del esquema E/R	73
5.1.1	Construcción del modelo conceptual.....	74
5.1.2	Supuestos semánticos específicos	82
5.1.3	Diagrama E/R.....	83
5.1.4	Semántica no reflejada en el diagrama E/R.	85
5.2	Diseño lógico estándar: obtención del esquema relacional	85
5.2.1	Transformación del esquema conceptual al esquema relacional.....	85
5.2.1.1	Transformación de entidades.....	86
5.2.1.2	Transformación de los atributos	86
5.2.1.3	Transformación de dependencias en identificación.....	88
5.2.1.4	Transformación de tipo y subtipos.....	90
5.2.2	Diagrama relacional.....	91
5.3	Semántica no reflejada en el diagrama relacional	92
5.3.1	Comprobaciones derivadas de los dominios de los atributos.....	92
5.3.2	Semántica perdida en la transformación del diagrama E/R al relacional.....	93
5.3.2.1	Transformación de jerarquías	93
5.3.2.2	Cardinalidad mínima.....	94
6	DISEÑO DE LA APLICACIÓN WEB	95
6.1	Funcionalidades de la aplicación Web	95
6.2	Perfiles de usuario	99
6.3	Navegabilidad de la aplicación Web	102
6.4	Interfaz de la aplicación Web.....	105
6.4.1	Distribución de la pantalla	105
6.4.2	Guía de estilo empleada	109
6.4.2.1	Colores de la aplicación Web	109
6.4.2.2	Tipografía	111
6.5	Mecanismos de seguridad	112
6.5.1	Control de accesos	112
6.5.2	Prevención de <i>SQL injection</i>	114

7	DISEÑO ACCESIBLE DE LA APLICACIÓN WEB.....	117
7.1	Introducción a las pautas para un diseño accesible	118
7.2	Pautas de Accesibilidad al Contenido en la Web.....	120
7.2.1	Pauta 1.....	120
7.2.2	Pauta 2.....	122
7.2.3	Pauta 3.....	124
7.2.4	Pauta 4.....	128
7.2.5	Pauta 5.....	130
7.2.6	Pauta 6.....	133
7.2.7	Pauta 7.....	143
7.2.8	Pauta 8.....	143
7.2.9	Pauta 9.....	144
7.2.10	Pauta 10.....	146
7.2.11	Pauta 11.....	150
7.2.12	Pauta 12.....	152
7.2.13	Pauta 13.....	154
7.2.14	Pauta 14.....	157
7.3	Validación de la accesibilidad.....	158
7.3.1	Validación mediante TAW.....	158
7.3.2	Validación mediante herramienta HERA	160
7.3.3	Validación de HTML.....	162
7.3.4	Validación de CSS.....	162
7.3.5	Uso de diferentes navegadores	162
8	PLANIFICACIÓN FINAL Y ANÁLISIS DE COSTES	164
8.1	Planificación final.....	164
8.2	Análisis de costes	167
9	CONCLUSIONES Y TRABAJOS FUTUROS.....	170
9.1	Conclusiones finales	170
9.2	Trabajos futuros.....	171
10	GLOSARIO DE TÉRMINOS	173
11	BIBLIOGRAFÍA.....	176
1.1	Impresa	176
1.2	En formato electrónico	177
ANEXO A: CUESTIONARIO SOBRE LA BASE DE DATOS		180
ANEXO B: CUESTIONARIO SOBRE LA APLICACIÓN WEB.....		188
ANEXO C: INSTALACIÓN DE LA BASE DE DATOS Y DE LA APLICACIÓN WEB DE FEDACE.....		193
1.	Requisitos previos	193
2.	Instalar MySQL.....	194
3.	Instalar el servidor Apache	200
4.	Configurar el servidor Apache.....	202
5.	Instalar el módulo de PHP	203
6.	Instalar phpMyAdmin	206

7.	Cargar la base de datos de FEDACE en MySQL	207
8.	Instalar la aplicación Web	209
9.	Obtener un dominio gratuito en Internet	209
9.1	Registro y selección del dominio	209
9.2	Asociar la IP dinámica.....	210
 ANEXO D: COPIAS DE SEGURIDAD DE LA BASE DE DATOS...		212
1	Hacer una copia de seguridad	212
2	Recuperar una copia de seguridad.....	214

ÍNDICE DE FIGURAS

Figura 2-1 Salarios del equipo.....	19
Figura 2-2 Descomposición de tareas (WBS)	20
Figura 2-3 Diagrama PERT.....	22
Figura 2-4 Planificación inicial	24
Figura 2-5 Diagrama GANTT inicial	25
Figura 2-6 Distribución inicial del equipo y su coste entre las diferentes etapas.....	26
Figura 3-1: Símbolos de accesibilidad Web validada.....	39
Figura 3-2 Funcionamiento de Web dinámica	43
Figura 5-1 Dominio de los atributos de la entidad <i>RECURSO</i>	75
Figura 5-2 Dominio de los atributos de la entidad <i>ASOCIACIÓN</i>	75
Figura 5-3 Dominio de los atributos de la entidad <i>CENTRO</i>	76
Figura 5-4 Particularización de la entidad <i>RECURSO</i>	77
Figura 5-5 Dominio de los atributos de la entidad <i>SEDE</i>	79
Figura 5-6 Relación <i>consta_de</i> entre <i>RECURSO</i> y <i>SEDE</i>	79
Figura 5-7 Dominio de los atributos de la entidad <i>CONTACTO</i>	80
Figura 5-8 Relación <i>atendida_por</i> entre <i>SEDE</i> y <i>CONTACTO</i>	81
Figura 5-9 Dominio de los atributos de la entidad <i>COMENTARIO</i>	81
Figura 5-10 Relación <i>tiene</i> entre <i>SEDE</i> y <i>COMENTARIO</i>	82
Figura 5-11 Diagrama E/R	84
Figura 5-12 Claves principales	86
Figura 5-13 Claves alternativas	87
Figura 5-14 Transf. de los atributos multivaluados <i>CENTRO DE ATENCIÓN</i>	88
Figura 5-15 Transformación de los atributos multivaluados de la entidad <i>SEDE</i>	88
Figura 5-16 Transformación de la dependencia en identificación de <i>SEDE</i>	89
Figura 5-17 Transformación de la dependencia en identificación de <i>CONTACTO</i>	89
Figura 5-18 Transformación de la dependencia en identificación de <i>COMENTARIO</i> ...	89
Figura 5-19 Transformación subtipos <i>ASOCIACIÓN</i> y <i>CENTRO DE ATENCIÓN</i>	90
Figura 5-20 Diagrama relacional.....	91
Figura 6-1 Usuarios y permisos asociados.....	101
Figura 6-2 Diagrama de navegación de la aplicación Web.....	104
Figura 6-3 Barras de navegación principal y secundaria	106
Figura 6-4 Otros elementos de la interfaz	108
Figura 6-5 Página de identificación	108
Figura 6-6 Aplicación Web vista a través de escala de grises.....	110
Figura 6-7 Control de accesos a la aplicación Web.....	114
Figura 7-1 Tabla en HTML.....	132
Figura 7-2 Apariencia de la página Web sin hoja de estilos habilitada	135
Figura 7-3 Menús actualizados automáticamente	138
Figura 7-4 Menús actualizados manualmente	141
Figura 7-5 Barra de navegación principal con caracteres imprimibles entre enlaces...	150
Figura 7-6 Interfaz de TAW	159
Figura 7-7 Interfaz Web de HERA	161
Figura 8-1 Planificación final.....	165
Figura 8-2 Diagrama GANTT final.....	166

Figura 8-3 Salario ordinario y extraordinario del equipo	167
Figura 8-4 Distribución final del equipo y su coste en las diferentes etapas	168
Figura C-1 Instalación de MySQL. Paso 1	194
Figura C-2 Instalación de MySQL. Paso 2	194
Figura C-3 Instalación de MySQL. Paso 3	195
Figura C-4 Instalación de MySQL. Paso 4	195
Figura C-5 Instalación de MySQL. Paso 5	195
Figura C-6 Instalación de MySQL. Paso 6	196
Figura C-7 Instalación de MySQL. Paso 7	196
Figura C-8 Instalación de MySQL. Paso 8	196
Figura C-9 Instalación de MySQL. Paso 9	197
Figura C-10 Instalación de MySQL. Paso 10.....	197
Figura C-11 Instalación de MySQL. Paso 11.....	197
Figura C-12 Instalación de MySQL. Paso 12.....	198
Figura C-13 Instalación de MySQL. Paso 13.....	198
Figura C-14 Instalación de MySQL. Paso 14.....	198
Figura C-15 Instalación de MySQL. Paso 15.....	199
Figura C-16 Instalación de MySQL. Paso 16.....	199
Figura C-17 Instalación de MySQL. Paso 17.....	199
Figura C-18 Instalación de MySQL. Paso 18.....	200
Figura C-19 Instalación de Apache	201
Figura C-20 Página de bienvenida de Apache	202
Figura C-21 Página mostrada por <i>phpinfo()</i>	205
Figura C-22 Carga de la base de datos. Paso 1.....	207
Figura C-23 Carga de la base de datos. Paso 2.....	208
Figura C-24 Carga de la base de datos. Paso 3.....	208
Figura D 1 Hacer copia de seguridad. Paso 1.....	212
Figura D-2 Hacer copia de seguridad. Paso 2	213
Figura D-3 Hacer copia de seguridad. Paso 3	213
Figura D-4 Recuperar copia de seguridad. Paso 1.....	214
Figura D-5 Recuperar copia de seguridad. Paso 2.....	215
Figura D-6 Recuperar copia de seguridad. Paso 3.....	215

1 INTRODUCCIÓN

El completo desarrollo de una guía de recursos sobre daño cerebral adquirido, formada por una base de datos y una aplicación Web para su gestión, que cumpla con todos los criterios sobre accesibilidad para personas con alguna discapacidad, constituye el eje central del presente proyecto. En las siguientes páginas, se exponen la motivación y objetivos principales que persigue.

1.1 Motivación

Desde su nacimiento en la década de los 60 como un proyecto de propósito científico-militar en EEUU (Estados Unidos), pasando por su creciente implantación a principios y mediados de los años 90, Internet se ha convertido hoy día en una herramienta indispensable para gran cantidad de actividades humanas [1]. Es muy posible que nuestra visión del trabajo, del ocio, del estudio en las escuelas o universidades, de los medios de comunicación e incluso de las relaciones sociales fuera muy diferente a como lo es actualmente sin la aparición de Internet.

Sin embargo, esta herramienta que parece querer romper definitivamente todas las barreras existentes hasta ahora en la comunicación, puede caer en la paradoja de levantar nuevas a aquellos que precisamente más necesitan beneficiarse de esta comunicación universal: las personas con alguna discapacidad. Una nueva barrera, en esta ocasión no física sino tecnológica, puede apartar una vez más a un considerable número de personas de su plena integración. A pesar del enorme potencial que Internet posee en este sentido, pocas veces se piensa en la accesibilidad para personas con discapacidad a la hora de construir un nuevo sitio Web. Y es tan grande la relevancia que está adquiriendo la Web que, más tarde o más temprano, la persona que no sepa o no pueda hacer un uso completo de ella será imposible que pueda contar con las mismas posibilidades de desarrollo que una persona que sí la utilice.

La Federación Española de Daño Cerebral (FEDACE) es especialmente sensible a esta circunstancia, debido a que reúne en su organización a asociaciones y familiares de afectados por daño cerebral que presentan, en mayor o menor medida, algún tipo de discapacidad. Es por ello que FEDACE deseaba que la futura aplicación Web de gestión

de la guía de recursos sobre daño cerebral adquirido que se iba a desarrollar para sus asociados contase desde el primer momento con un diseño que cumpliera con todos los criterios existentes hoy en día en cuanto a accesibilidad Web.

1.2 Objetivos

Expuesto lo anterior, los objetivos que persigue el presente proyecto se pueden resumir en los siguientes:

- **Desarrollo completo de la base de datos.**

La construcción y puesta en funcionamiento de una base de datos que recoja los diferentes recursos existentes en España sobre el tratamiento del daño cerebral, debe hacerse siguiendo las distintas etapas que un proceso como éste requiere.

Primeramente, la toma de requisitos con el cliente debe llevarse a cabo utilizando todos aquellas técnicas que se consideren necesarias, teniendo en cuenta que se parte desde una situación totalmente inicial, sin trabajo previo alguno, y en la que el cliente todavía no tiene una visión clara de lo que está buscando. El análisis y el diseño deberán ser minuciosos y elaborados, puesto que desde esta etapa se pueden localizar y subsanar posibles errores y carencias en etapas posteriores. Una vez refinado el diseño, ha de ser explicado al cliente de la forma más clara posible para certificar que se cumple con todos los requisitos marcados y expectativas. La implementación, bajo la herramienta software escogida, será fiel al guión dictado por los esquemas y diagramas obtenidos en las etapas previas. Como fase previa a la instalación, una correcta y exhaustiva batería de pruebas debe asegurar el correcto funcionamiento de todo el sistema bajo todas las posibles situaciones que se encontrará en su uso.

- **Construcción de una aplicación Web totalmente accesible para la gestión de la base de datos.**

Para poder administrar y consultar la base de datos que sustenta la guía de recursos sobre daño cerebral, se desarrollará una aplicación Web a la que se tendrá acceso tanto desde dentro del lugar de trabajo del cliente como desde el exterior por cualquier persona con conexión a Internet.

Durante todo el proceso de creación de la aplicación Web se pondrá muy especial atención en cumplir con todos los requisitos existentes en materia de accesibilidad, siguiendo los estándares disponibles, utilizando las herramientas de verificación escogidas y certificando el cumplimiento del nivel de conformidad accesible más exigente, conocido como AAA [2]. La documentación recogida tanto en el W3C [26] (siglas en inglés de *World Wide Web Consortium*) como en el WAI [3] (siglas de *Web Accessibility Initiative*, Iniciativa de Accesibilidad Web) será la constante referencia con respecto a este tema. El análisis de las funcionalidades necesarias en dicha Web y su diseño se verán muy influidos por los criterios de accesibilidad, dando siempre prioridad a una aplicación Web accesible y funcional que una innovadora o de complejo diseño, pero vetada a personas con alguna discapacidad. Igualmente, se buscará la completa conformidad del marcado HTML (*Hyper Text Markup Language* o Lenguaje de Marcación de Hipertexto) utilizado y las hojas de estilo en cascada o CSS (*Cascading Style Sheets*) empleadas con los estándares definidos para ambos en el W3C.

– **Instalación completa del sistema.**

Tanto la base de datos con la guía de recursos como la aplicación Web de acceso a la misma deben ser instaladas de forma completa en las instalaciones desde las cuales se va a hacer uso de ellas. La instalación, además de asegurar una explotación correcta y adecuada del sistema por parte del cliente, debe ir acompañada del software necesario para realizar futuras instalaciones y de la correspondiente documentación generada, incluyendo tanto la documentación técnica como la documentación destinada al usuario final.

– **Estudio y elección del software más adecuado.**

Dado el alcance que toma el presente proyecto, se puede prever que se necesitarán diferentes clases de software y herramientas durante todo el proceso de su desarrollo. Por ello, es fundamental saber escoger aquello que asegure un óptimo funcionamiento y una adecuación correcta a las necesidades y requisitos marcados.

En cuanto a la base de datos, se hace necesario estudiar la amplia gama de Sistemas Gestores de Bases de Datos (SGBD) existentes hoy día y analizar las ventajas e inconvenientes de cada uno de ellos, para decantarse por aquél que mejor se ajuste a la situación planteada.

La aplicación Web, por otro lado, deberá ser alojada en un servidor propio instalado en el lugar de trabajo del cliente, lo que conlleva igualmente a analizar las diferentes opciones que dentro del mundo de los servidores HTTP (*Hypertext Transfer Protocol* o Protocolo de Transferencia de Hipertexto) existe. La versión final de la aplicación Web implicará así mismo una conexión frecuente con la base de datos para recuperar o almacenar información, de modo que se deberá también buscar aquella tecnología dentro de los lenguajes de programación que mejores resultados proporcione en este sentido.

– **Coste mínimo asociado.**

El resultado final de cualquier proyecto siempre queda determinado en gran medida por el presupuesto del que se dispone para su realización. En el caso que nos ocupa y tratándose de una organización sin ánimo de lucro, el desarrollo completo de la guía de recursos sobre daño cerebral adquirido deberá hacerse siempre mirando por aquel software que menor coste represente, ya que no se desea pagar ni grandes licencias de explotación ni costosos manuales de uso.

1.3 Estructura del documento

El presente documento se estructura de la siguiente forma:

En el capítulo 1, se explica la introducción al proyecto, su motivación y los objetivos básicos que persigue la realización del mismo.

El capítulo 2 expone la gestión inicial del proyecto, donde se explica la metodología seguida, el ciclo de vida escogido para el proyecto, la organización del trabajo con el correspondiente diagrama WBS (siglas de *Work Breakdown Structure*), el diagrama PERT (*Program Evaluation and Review Technique* o Técnica de Revisión y Evaluación de Programas) y GANTT, así como la planificación temporal.

El capítulo 3 aborda el estado del arte, indicando cual es la situación actual y los antecedentes de aquellos aspectos íntimamente ligados con el presente proyecto: historia

y objetivos de FEDACE, accesibilidad Web y trabajos hechos en este campo, sistemas gestores de bases de datos, servidores HTTP y otro software y tecnologías relacionados.

En el capítulo 4 se explican las reuniones mantenidas con el cliente, la toma de requisitos y el análisis exhaustivo de los mismos para obtener una definición general del sistema a desarrollar. El capítulo 5 trata el diseño completo de la base de datos, incluyendo el modelo conceptual y el modelo lógico. El diseño de la aplicación Web, sus funcionalidades, tipo de interfaz y mecanismos de seguridad se detallan en el capítulo 6.

El capítulo 7 está dedicado por completo a la accesibilidad de la aplicación Web. Se detallan los diseños propuestos para lograr la plena accesibilidad siguiendo siempre las recomendaciones marcadas desde el W3C.

En el capítulo 8 se realiza la planificación final y el análisis de costes comparándola con la planificación inicial descrita en el capítulo 2, así como los costes materiales y humanos para poder llevarlo a cabo. En el capítulo 9 se hace un repaso a la repercusión del proyecto, conclusiones obtenidas a través de la realización del mismo y posibles mejoras o trabajos futuros relacionado. Los capítulos 10 y 11 detallan, respectivamente, el glosario de términos empleados y la bibliografía utilizada en el presente documento.

Por último, se ha anexo al documento los cuestionarios utilizados en la toma de requisitos, el manual de instalación completa de la guía de recursos sobre daño cerebral adquirido y el manual sobre realización y recuperación de copias de seguridad de la base de datos.

2 GESTIÓN DEL PROYECTO

Dentro del apartado de gestión de un proyecto se engloban todas aquellas actividades destinadas a asegurar la correcta realización del mismo, exponiendo el modelo de desarrollo que se va a seguir, enumerando los recursos humanos y materiales necesarios, planificando en el tiempo las diferentes tareas y distribuyendo trabajos y responsabilidades entre los diferentes miembros que componen el equipo de trabajo.

2.1 Gestión del software

A continuación se exponen la metodología y el ciclo de vida escogidos para la elaboración y desarrollo del presente proyecto, así como las herramientas utilizadas.

2.1.1 Metodología

La metodología es un conjunto de técnicas y herramientas que orientan el desarrollo del software, proporcionando una visión global del proyecto a las personas encargadas de llevarlo a cabo.

Con respecto a la base de datos, se ha optado una de las metodologías más utilizadas en el desarrollo de este tipo de sistemas [“Diseño de bases de datos relacionales”, 1999]:

- Toma de requisitos
- Análisis de requisitos
- Modelado conceptual: obtención del esquema entidad-interrelación (E/R)
- Diseño Lógico Estándar: obtención del esquema relacional
- Diseño Lógico Específico

En cuanto al desarrollo de la aplicación Web, se trata de obtener una herramienta gráfica destinada a la gestión de la base de datos. No se busca ninguna aplicación compleja o con un largo y elaborado proceso de desarrollo, sino una interfaz básica y sencilla. Así, la metodología aplicada será la siguiente [Diseño de páginas Web, 1997]:

- Toma de requisitos
- Análisis de requisitos
- Diseño de interfaces
- Diseño de la navegabilidad Web
- Implementación de plantillas HTML y CSS
- Pruebas.

2.1.2 Ciclo de vida

El ciclo de vida determina las fases por las que un desarrollo de software debería pasar, así como su orden, para la finalización exitosa del proyecto y la consecución de sus objetivos. La definición de un ciclo de vida además facilita enormemente el control sobre los tiempos en los que es necesario aplicar los diferentes recursos.

El ciclo de vida escogido para este proyecto es la entrega evolutiva, mediante el cual se diseñan y desarrollan diferentes versiones cada vez más ampliadas del sistema requerido. Cada entrega de una versión supone la toma de algunos nuevos requisitos, matización de otros e incluso la supresión de aquellos que ya no se consideren necesarios por parte del cliente. Fomenta la constante comunicación entre el desarrollador y el usuario final, permitiendo una mayor satisfacción y un cubrimiento más efectivo de las necesidades del cliente.

Durante la elaboración del presente proyecto, se entregará una versión inicial y básica, con las funcionalidades más urgentes, de manera que el cliente pueda comenzar a usarla cuanto antes en sus tareas diarias y le sirva para perfilar nuevos requisitos y expresar su conformidad o no con los ya implementados. Posteriormente, tras nuevas reuniones con el cliente y diseños más exhaustivos por parte de los desarrolladores si fuese necesario, se actualizará esta versión inicial con una o varias hasta llegar a la versión definitiva, meta que, evidentemente, marcará el cliente.

2.1.3 Recursos software y hardware

Los recursos software utilizados, en cada una de las etapas, serán los siguientes:

- **Toma de requisitos, análisis y diseño**
 - Microsoft Word 2002: procesador de textos usado para la redacción de los requisitos y explicación del diseño propuesto.
 - Microsoft Visio 2003: herramienta gráfica utilizada para la realización de diagramas y esquemas como el esquema entidad-interrelación (ER), el esquema relacional o el WBS.
 - Microsoft Project 2003: gestor de proyectos utilizado para la planificación y la obtención del diagrama GANTT.
- **Implementación y pruebas**
 - MySQL 5.0.18: será el SGBD escogido para almacenar la BD.
 - Servidor Apache 2.0: servidor de Internet escogido para la publicación de la aplicación Web de acceso a la base de datos.
 - EditPlus 2.0: editor de texto utilizado para la parte de implementación en los diferentes lenguajes de programación usados en este proyecto.
 - HTML 4.01 *Transitional*: lenguaje de marcado de páginas de hipertexto, para la construcción de las diferentes páginas que compondrán la aplicación Web.
 - CSS: para la maquetación de la Web.
 - PHP 5.0: siglas en inglés de *Hypertext Preprocessor*, lenguaje del lado del servidor para integrar con páginas HTML y permitir la conexión con la BD.
 - Validador TAW 3.08: herramienta de validación automática de accesibilidad Web.
 - Validador HERA 2.0: validador automático de accesibilidad.
- **Documentación**
 - Microsoft Word 2002: para la redacción del manual de usuario, manual técnico y la presente memoria del proyecto.
 - Microsoft Visio 2003: realización de varios diagramas y esquemas.
 - Microsoft Power Point 2002: herramienta gráfica para el diseño de algunas figuras presentes en esta memoria.
 - Microsoft Project 2003: para la obtención de diagramas de seguimiento y planificación.

Por otro lado, los recursos hardware utilizados en la elaboración del proyecto son:

- Ordenador portátil: equipo para la realización de la mayor parte de las actividades de implementación y diseño, útil especialmente para las reuniones con el cliente y la muestra de elementos de prueba. Su modelo es HP Pavilion, procesador Intel de 3 MHz, 512 MB de RAM y sistema operativo Windows XP.
- Ordenador de sobremesa: equipo destinado a las pruebas de instalación y publicación de la Web mediante servidor, con conexión permanente a Internet. El modelo es HP Pavilion, procesador AMD de 1MHz, 1 GB de memoria RAM y sistema operativo Windows XP.
- Conexión a Internet de 20 MB/s de ancho de banda.
- Impresora HP LaserJet

2.1.4 Recursos humanos

El equipo humano necesario para la elaboración del proyecto se ha considerado el siguiente:

- Jefe de Proyecto: encargado principal de dirigir el proyecto, de identificar riesgos, aplicar planes de contingencia, redirigir esfuerzos en la dirección correcta y coordinar a los demás miembros del equipo. Su perfil debería ser el de una persona con experiencia en la dirección de otros proyectos.
- Analista de bases de datos: su misión es la de tomar los requisitos de usuario con respecto a la base de datos del sistema y analizarlos para identificar y acotar las necesidades expresadas por el cliente
- Analista de la aplicación Web: el cometido del analista de la aplicación Web es la de tomar los requisitos de usuario con respecto a ésta y definir un producto Web que se ajuste a los deseos manifestados en las reuniones con el cliente.
- Diseñador de la base de datos: en sus manos estará la tarea de diseñar los modelos conceptual y lógico de la base de datos, en función de los detalles proporcionados por el analista.
- Diseñador de la aplicación Web: encargado de realizar un diseño de la aplicación Web de acceso a la base de datos, prestando especial cuidado en la accesibilidad y utilizando para ellos los resultados de los análisis suministrados

por el analista de la aplicación Web y por la documentación existente sobre accesibilidad Web.

- Programador de la base de datos: implementará el diseño realizado sobre la base de datos, utilizando para ello la sintaxis del Lenguaje de Consulta Estructurado o SQL (*Structure Query Language*) utilizada por el SGBD escogido. El mantenimiento de la base de datos y la instalación del software son también sus responsabilidades. Su conocimiento en SQL estándar y administración de bases de datos debería ser amplia.
- Programador de la aplicación Web: su trabajo consiste en implementar mediante HTML y CSS los diseños realizados sobre la Web de acceso a la base de datos, poniendo énfasis en el diseño accesible. A su cargo también está la conexión a la base de datos mediante el lenguaje PHP, en el que debería tener sólida experiencia. Así mismo, deberá instalar y mantener el servidor Web y todo el software relacionado.

A pesar de que este equipo sería el recomendado para la realización del proyecto, realmente todas y cada una de las actividades en su desarrollo serán llevadas a cabo por una única persona, el autor del presente proyecto, que desempeñará según los casos un papel u otro.

El sueldo calculado para cada uno de los miembros del equipo, basándose de forma aproximada en los existentes hoy en día en el mercado [46], será el mostrado en la Figura 2-1:

Cargo	Salario
Jefe de Proyecto	25 €/hora
Analista de BD	20 €/hora
Analista Web	20 €/hora
Diseñador de BD	20 €/hora
Diseñador Web	20 €/hora
Programador de BD	15 €/hora
Programador Web	15 €/hora

Figura 2-1 Salarios del equipo

2.2 Organización del trabajo

El WBS es un diagrama con forma de árbol donde se representan cada una de las tareas y subtareas en las que está dividido un proyecto. El WBS de este proyecto, expandido hasta un tercer nivel, es el mostrado en la siguiente figura (Figura 2-2). La numeración de las tareas y subtareas será la utilizada en el posterior diagrama PERT (Figura 2-3):

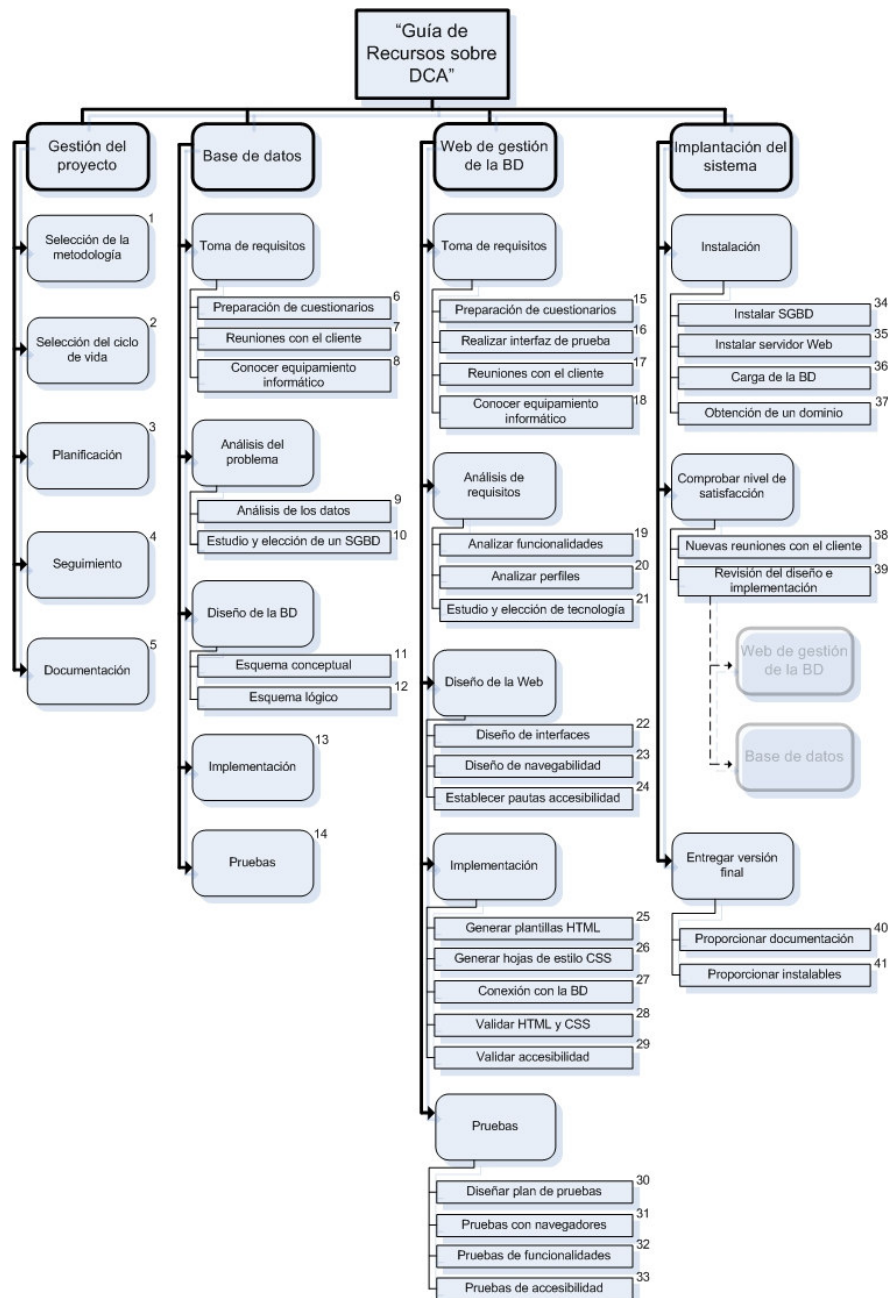


Figura 2-2 Descomposición de tareas (WBS)

2.3 Planificación inicial

La planificación inicial se realiza en las etapas inmediatamente posteriores al comienzo del proyecto y pretende dar una visión general de las diferentes actividades repartidas a lo largo del tiempo. Mediante la planificación se estiman los recursos que se van a asignar a cada una de las etapas del desarrollo y la duración de éstas, sirviendo como guía para conocer en cada momento en qué nivel de desarrollo se encuentra el proyecto.

2.3.1 Diagrama PERT de actividades

El diagrama PERT constituye una popular herramienta gráfica que facilita la administración y gestión de todo tipo de proyectos. Es, básicamente, un método para analizar las tareas involucradas en completar un proyecto dado, especialmente el tiempo para completar cada tarea, e identificar el tiempo mínimo necesario para completar el proyecto total. Su invención se dio en EEUU en la década de 1950, durante la elaboración de un proyecto de naturaleza militar.

En la siguiente hoja se muestra el diagrama PERT correspondiente al presente proyecto (Figura 2-3). Cada tarea está representada con un rectángulo con una numeración en el centro que indica la tarea concreta a la que hace referencia. La numeración utilizada es la misma que en el diagrama GANTT (Figura 2-2). Así mismo, cada tarea indica también su duración en días, la fecha más temprana y más tardía de comienzo y la fecha más temprana y tardía de finalización. Al ser el ciclo de vida escogido entrega evolutiva, existen ciertas etapas o fases del proyecto por las que se volverá a pasar en cada desarrollo de una nueva versión. La planificación inicial establece la creación de dos versiones, por lo que las etapas que se repiten aparecen una única vez en el diagrama PERT y la duración estimada para éstas es la suma de las duraciones estimadas en cada una de sus dos iteraciones. El resultado final establece una duración aproximada de 94 días laborables para la consecución del proyecto.

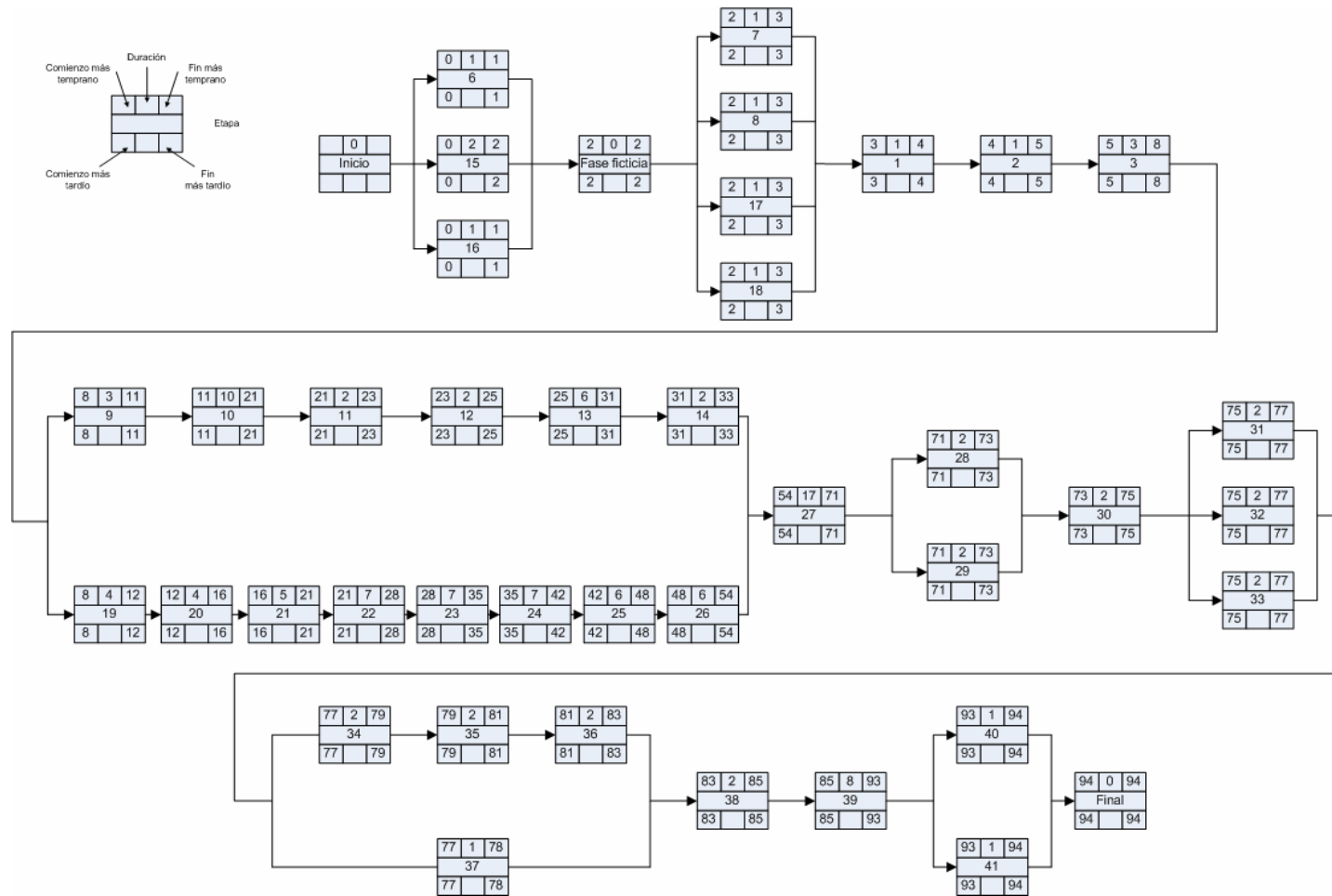


Figura 2-3 Diagrama PERT

2.3.2 Diagrama GANTT

El diagrama GANTT (Figura 2-5) sitúa de forma secuencial las diferentes etapas de un proyecto y las tareas que las componen a lo largo de una línea temporal que representa el tiempo total de desarrollo del proyecto.

Así mismo, distribuye los recursos humanos entre cada tarea siendo el periodo laboral establecido de la siguiente manera:

- Periodo laboral de lunes a viernes
- Jornada diaria de 8 horas
- Sábados y domingos festivos
- Los siguientes días también se considerarán festivos: en el mes de Diciembre de 2005, los días 5, 6 y 8; en el mes de Enero del año 2006, el día 1; en el mes de Marzo, serán festivos los días 13 y 14; en el mes de Mayo, los días 1, 2 y 15.

A continuación se muestran la planificación prevista para el proyecto (Figura 2-4) y el correspondiente diagrama GANTT (Figura 2-5). La planificación muestra de forma más detallada las tareas expuestas en el diagrama WBS (Figura 2-2), así como su distribución temporal. La primera columna asigna un número identificador a cada tarea siguiendo un orden cronológico. La segunda de las columnas de la planificación describe la tarea o subtarea a la que se hace referencia. La tercera columna señala en días laborales cuánto tiempo se estima que dura la tarea en cuestión. Las dos siguientes columnas indican la fecha exacta de inicio y de finalización de la tarea. La sexta columna muestra aquellas tareas que preceden a la tarea descrita. Por último, se acompaña a la planificación de una columna adicional donde se detallan los recursos humanos que se estima emplear en cada una de las tareas. Cada miembro del equipo va acompañado de un porcentaje que indica la disponibilidad del trabajador para esa tarea, tomando como base las 8 horas diarias de jornada laboral.

Figura 2-4 Planificación inicial

Figura 2-5 Diagrama GANTT inicial

El tiempo total invertido se prevé de 722,8 horas y un gasto derivado total de 13717 euros. La distribución del tiempo y del trabajo de los distintos miembros del equipo de desarrollo, incluyendo los costes asociados, quedan reflejados en la siguiente figura (Figura 2-4):

Etapas	Duración (horas)	Recursos		Coste (€)
		Empleado	Horas	
Toma de requisitos	69	Jefe del Proyecto	24,60	1463
		Analista BD	14,20	
		Analista Web	14,20	
		Diseñador Web	8	
		Programador Web	8	
Análisis de requisitos	128	Analista BD	56	2560
		Analista Web	72	
Diseño global	139,4	Diseñador BD	30	2788
		Diseñador Web	109,4	
Implementación versión 1.0	164	Diseñador Web	16	2690
		Programador BD	30	
		Programador Web	118	
Pruebas versión 1.0	28	Diseñador BD	4	560
		Diseñador Web	12	
		Programador Web	12	
Instalación versión 1.0	7,2	Jefe de Proyecto	2	128
		Programador BD	2,6	
		Programador Web	2,6	
Comprobar nivel de satisfacción	56	Jefe de Proyecto	16	1200
		Analista BD	8	
		Analista Web	8	
		Diseñador BD	12	
		Diseñador Web	12	
Implementación versión 2.0	60	Programador BD	4	900
		Programador Web	56	
Pruebas versión 2.0	32	Diseñador BD	4	580
		Diseñador Web	16	
		Programador Web	12	
Instalación versión 2.0	7,2	Jefe de Proyecto	2	128
		Programador BD	2,6	
		Programador Web	2,6	
Comprobar nivel de satisfacción	24	Jefe de Proyecto	8	520
		Analista BD	4	
		Analista Web	4	
		Diseñador BD	4	
		Diseñador Web	4	
Entregar versión final	8	Jefe de Proyecto	8	200
TOTAL HORAS	722,8	TOTAL COSTE		13717

Figura 2-6 Distribución inicial del equipo y su coste entre las diferentes etapas

3 ESTADO DEL ARTE

3.1 Un caso real: desarrollo de una Web accesible

Voluntariado es aquella actividad solidaria, altruista y no remunerada, que realiza una persona por elección propia, dedicando para ello una parte de su tiempo y esfuerzo. Actualmente existe en España un importante movimiento de voluntarios, especialmente entre los más jóvenes, que por motivos políticos, religiosos, morales o sencillamente por la necesidad de sentirse socialmente útiles, acuden a asociaciones u organizaciones para formar parte en alguna actividad de voluntariado. Internet, evidentemente, no ha quedado al margen de este movimiento social [32].

El portal de voluntariado Haces Falta tiene como objetivo facilitar y fomentar el voluntariado, utilizando Internet para poner en contacto a aquellas personas que desean invertir parte de su tiempo y trabajo en colaborar de forma voluntaria con organizaciones no lucrativas que necesitan y buscan ayuda. Igualmente, mediante publicaciones, foros, mensajes y noticias persiguen sensibilizar a la sociedad sobre la importancia del voluntariado [4].

Precisamente fue desde esta Web desde donde FEDACE solicitaba a una persona voluntaria, preferiblemente con estudios informáticos o experiencia en este campo, para crear una base de datos donde poder almacenar y consultar información relacionada con sus recursos (asociaciones españolas federadas y centros de atención de todo tipo).

FEDACE está formada por una veintena de asociaciones españolas de afectados y familiares de afectados por una lesión cerebral. Sus actividades, entre otras, son coordinar a las diferentes asociaciones mediante congresos y publicaciones, representarlas en sus relaciones con la Administración, realizar campañas de sensibilización y prevención, proporcionar información y asesoramiento a las familias y fomentar la investigación acerca del daño cerebral. Uno de los principales servicios que oferta FEDACE es la disponibilidad de un elaborado listado de diferentes recursos disponibles tanto para los afectados directos por el daño cerebral como para sus familiares, incluyendo hospitales, centros de rehabilitación, residencias, centro de ocio y tiempo libre, centros de formación e inserción laboral, otras asociaciones, etc. [5].

Hasta ahora, FEDACE había recopilado toda esta información en un documento de texto, a modo de publicación, que enviaba cada cierto tiempo a las diferentes asociaciones y entidades vinculadas. Con el paso del tiempo, y según fueron aumentando los recursos presentes en dicho informe, la actualización de los recursos ya existentes así como el proceso de obtención de datos de los nuevos recursos que iban apareciendo se hacía más tediosa y costosa. El tiempo que se tardaba en confirmar todos los datos de contacto y servicios ofrecidos por los recursos publicados en ediciones anteriores, más el tiempo empleado en ponerse en contacto con los nuevos recursos con la intención de recavar toda la información posible sobre ellos, hacía cada vez más difícil disponer de un listado periódico de recursos sobre daño cerebral verdaderamente útil y fiable para el conjunto de asociaciones de la federación. La gran heterogeneidad de los recursos en cuanto a los servicios ofrecidos, modelos de atención, titularidad (públicos, privados, privados concertados, etc.) o profesionales de los que disponen hacían difícil poder ajustar el recurso a las necesidades concretas de la persona con daño cerebral a la que se quería ayudar. Teniendo en cuenta, además, que las consecuencias del daño cerebral son múltiples y variadas y pueden afectar de muy diferentes formas al enfermo, necesitando éste una atención y servicios muy concretos según las dolencias acusadas.

Desde la primera reunión con FEDACE se constató la intención no sólo de disponer de una base de datos desde la cual automatizar el proceso de recogida y consulta de los recursos sobre daño cerebral disponibles, sino también de darle un enfoque más práctico y aprovechar de alguna forma la Web ya existente de la Federación para ofrecer de forma cómoda y más amplia esta información a cualquier persona interesada en recibirla. Es por ello que se propuso la creación de una base de datos que mediante una aplicación Web, vinculada a la que ya se disponía, pudiera ser tanto administrada como consultada. El proceso de desarrollo no abarcaba ya únicamente la implementación de la base de datos, sino que también sería necesario desarrollar una Web, instalar un servidor y realizar una selección del software y de las técnicas más apropiadas y que mejor se ajustasen al perfil final del cliente, es decir, FEDACE. El desarrollo completo de todo el sistema (base de datos, servidor, Web) se convierte entonces en un proyecto amplio que involucra muchas áreas de la ciencia informática.

La construcción de la base de datos comienza desde cero, no existe ninguna toma de requisitos previa, una base de datos ya existente e informatizada de la que partir ni una idea muy concreta de lo que se está buscando. Las reuniones con el cliente, la elaboración de cuestionarios, la presentación y explicación de prototipos y el consecuente refinamiento de los mismos se hacen una constante. Se ha de elegir entre todas la gama disponible un SGBD adecuado a las características de los equipos informáticos presentes y que además sea económico.

Por otro lado, la página Web de la que dispone FEDACE fue creada por una empresa privada externa y es actualmente mantenida por la misma. Su único acercamiento a la accesibilidad consiste en una versión en texto plano de la misma página Web, opción poco recomendada y que es siempre advertida como el último recurso al que acudir en el caso de ser imposible la transformación accesible de la página original. Tratándose de una asociación de personas que presentan muy diversos y variados tipos de discapacidad y que la aplicación va principalmente dirigida a ellas, se presenta como algo imperativo tener siempre presente las pautas de accesibilidad Web durante todas las etapas de desarrollo de la misma.

Comenzaba así un proceso de desarrollo amplio y que afectaba a diversos campos de la informática, donde se tuvo que conocer y estudiar las diferentes opciones existentes en cuanto a software y lenguajes de programación, sus características, ventajas e inconvenientes, selección de los más idóneos para el caso y obtención de una extensa documentación que permitiera un completo aprendizaje sobre ellos. Sin olvidar, en el caso de la aplicación Web, buscar aplicar en todo momento los criterios y pautas para cumplir con los niveles de accesibilidad que se espera de una aplicación especialmente dirigida a aquellos que, hasta el día de hoy, se han ido encontrando una vez más apartados y discriminados de una tecnología como Internet que podría ser la definitiva impulsora de su tan ansiada plena integración.

3.2 Federación Española de Daño Cerebral (FEDACE)

Daño Cerebral Adquirido.

El daño cerebral es una lesión que se produce en las estructuras cerebrales de forma súbita en personas que, habiendo nacido sin ningún tipo de daño en el cerebro, sufren en un momento posterior de su vida, lesiones en el mismo como consecuencia de un accidente o una enfermedad. Se utiliza, por tanto, la definición de lesión cerebral cuando hay un problema suficientemente importante que afecta a la cabeza, dañando no solamente la caja craneal sino también el cerebro. La lesión cerebral más frecuente es el traumatismo craneoencefálico (TCE) producido por accidentes de tráfico, laborales, deportivos, caídas o agresiones. Si bien existen otras muchas causas, como tumores cerebrales, anoxias cerebrales por infarto de miocardio, enfermedades metabólicas o accidentes cerebro-vasculares (ACV), más usuales entre la gente mayor que las lesiones traumáticas. Más del 80% de los afectados de traumatismo craneoencefálico se debe a accidentes de tráfico de jóvenes entre los 15 y 30 años [5].

El daño cerebral provoca generalmente, en primer lugar, un cambio repentino del estado de consciencia, cuya gravedad y duración son variables (desde una confusión mínima en tiempo y en contenido hasta un estado de coma profundo o coma vegetativo persistente en los casos más graves). Siempre se produce una alteración de las funciones cerebrales: perturbaciones en los comportamientos de la vida cotidiana (personal, familiar, académica, social, profesional). Y puede dar lugar a secuelas importantes en el plano neuropsicológico, que no están siempre en relación directa con el grado de afección constatado en el plano físico o neurológico. Incluso, aunque existen ciertas características comunes en las personas afectadas por una lesión cerebral, cada una debe ser considerada como un caso único que posee características propias.

A pesar de la magnitud del problema, existe una ignorancia casi absoluta acerca de la lesión cerebral. Perdura todavía la falsa creencia de que después del coma el afectado abre los ojos y regresa a su vida anterior al accidente. La realidad es otra bien distinta: el afectado y su familia tendrán que aprender a vivir casi desde cero. Los bulos o falsas creencias abundan, y el conocimiento real sobre qué es una lesión cerebral escasea. "A fuerza de voluntad se puede conseguir todo" es uno de los falsos lemas que atribuyen a la voluntad un poder que no tiene. Desafortunadamente, incluso la mejor voluntad del

mundo, no puede regenerar un cerebro dañado sin una ingente colaboración de especialistas expertos en el tratamiento de la lesión cerebral.

Historia de FEDACE y objetivos que persigue.

FEDACE, constituida el 13 de Octubre de 1995, está formada por asociaciones de familiares y de afectados por un Daño Cerebral Adquirido. En la actualidad cuenta con un total de 24 asociaciones, que agrupan a más de 6.000 familias. Nace con el objetivo de concienciar a la sociedad y a la Administración sobre la importancia de crear una infraestructura para tratar a los afectados, acorde con el número de personas que sufren cada año un daño cerebral adquirido.

FEDACE es miembro fundador de la "Confederación Europea de Lesionados Cerebrales y Familias" (en inglés *Brain Injured & Families European Confederation*, siglas BIF EC) formada por Reino Unido, Francia, Dinamarca, Bélgica, Italia, Suiza, Austria, Alemania, Irlanda, Países Bajos, Suecia, Eslovenia, Finlandia y Noruega. Esta Confederación se creó en Bruselas en abril de 1999, y a través de ella formamos parte del Foro Europeo de la Discapacidad (EDF) [5]. Así mismo, FEDACE pertenece al Comité Español de Representantes de Personas con Discapacidad (CERMI), *European Disability Forum* (EDF) y al Comité Español para el Bienestar Social (CEBS) [Despertar, 2005].

La Federación tiene una línea muy clara de actuación desde sus orígenes. Los objetivos por los que lucha y sigue trabajando son [5]:

- La defensa de los derechos de las personas con Lesión Cerebral.
- Concienciar a la sociedad y a la Administración de los problemas y las necesidades, tanto humanas como sociales, que sufren las personas con Lesión Cerebral.
- La prevención mediante el impulso y la colaboración en campañas y programas.
- El fomento de la creación de centros de rehabilitación y otros servicios que cubran las necesidades del lesionado cerebral desde su fase inicial hasta su plena integración socio laboral.
- El establecimiento de un protocolo de rehabilitación y de una guía para las buenas prácticas.

- La realización de estudios epidemiológicos del colectivo de Daño Cerebral.
- La promoción de líneas de investigación y desarrollo destinadas a lograr un conocimiento más profundo de la lesión cerebral.
- Fomentar la creación de nuevas asociaciones de Daño Cerebral en aquellas comunidades y provincias donde no existan.

Las asociaciones que componen FEDACE ofrecen ayuda a los afectados de Daño Cerebral y a sus familiares desde el primer momento. Los programas asistenciales que ponen a su disposición son:

- Programa de información, orientación y apoyo a los afectados de daño cerebral y a sus familias.
- Programa de centros de tratamientos: algunas asociaciones cuentan con sus propios centros de tratamientos en los que se lleva a cabo la rehabilitación integral del afectado, realizada por un equipo multidisciplinar: fisioterapia, logopedia, terapia ocupacional, rehabilitación neuropsicológica, etc.
- Programa de ayuda a domicilio.
- Talleres ocupacionales.
- Actividades de ocio y tiempo libre.
- Programa de grupos de auto-apoyo.

3.3 Centro Español de Subtitulado y Audiodescripción (CESyA)

Durante aproximadamente 6 meses, se estuvo compatibilizando el desarrollo de este proyecto con una beca de 5 horas diarias en CESyA [6], realizando labores de apoyo en el diseño e implementación de una base de datos, instalación y mantenimiento de un servidor Web y, lo mas importante, en el desarrollo de una Web totalmente accesible que realizara conexiones con la base de datos de CESyA.

CESyA es un centro dependiente del Real Patronato sobre Discapacidad, organismo a su vez dirigido desde el Ministerio de Trabajo y Asuntos Sociales, que persigue fomentar la accesibilidad en los medios audiovisuales españoles a través de las herramientas de subtitulado y audiodescripción, ayudar a la formación e investigación

de dichas técnicas y, en general, apoyar la sensibilización social y la creación de material audiovisual accesible en España. Para lograr su objetivo, CESyA colabora estrechamente con el Ministerio de Trabajo y Asuntos Sociales, con organizaciones de representantes de personas con alguna discapacidad y sus familias, productoras de subtítulo y audiodescripción, así como con diversos radiodifusores. De esta forma, se reúne a todos los agentes implicados en la accesibilidad audiovisual y sirve de foro para el diálogo y de plataforma de iniciativas para todos ellos.

La constitución de CESyA satisface una demanda de largos años por parte de las asociaciones y familiares de personas con alguna discapacidad en pos de lograr una verdadera accesibilidad universal. CESyA se ha incluido en los anteproyectos de la “Ley de Lengua de Signos y de Apoyo a los Sistemas de Comunicación Oral y de la Ley General Audiovisual” [7]. Entre las asociaciones que colaboran con el Centro, se encuentran el Comité Español de Persona con Discapacidad (CERMI), la Confederación Nacional de Sordos Españoles (CNSE), la Federación Española de Asociaciones de Padres y Amigos de los Sordos (FIAPAS), y la Organización Nacional de Sordos Españoles (ONCE) [6].

La Universidad Carlos III de Madrid fue elegida por concurso público en 2005 entre otras universidades españolas para poner en marcha el proyecto CESyA. La Universidad cuenta con un amplio historial a favor de la accesibilidad. Presume de amplio número de profesores y académicos que han ido especializándose en este campo, situando a la Universidad como una referencia en la enseñanza e investigación accesible no sólo dentro de España sino también fuera de ella. Ya sus estatutos, aprobados en 2002, recogían de forma destacada la especial atención a las personas con alguna discapacidad [8]:

Artículo 3

(...)

3. La Universidad promoverá la integración en la comunidad universitaria de las personas con discapacidades.

Artículo 96

Son derechos del personal docente e investigador, sin perjuicio de cualquier otro derecho o facultad reconocido en el ordenamiento jurídico:

(...)

b) Disponer de los medios necesarios para el cumplimiento de sus obligaciones, con atención específica a las personas con discapacidades y de acuerdo a las posibilidades con que cuente la Universidad.

Artículo 116

Son derechos de los estudiantes de la Universidad:

(...)

d) Disponer de unas instalaciones adecuadas que permitan el normal desarrollo de los estudios, con atención específica a las personas con discapacidades.

Esta situación de fomento de la accesibilidad ha permitido a la Universidad Carlos III disponer de las tecnologías y herramientas de más alto nivel para el desarrollo e investigación que el proyecto CESyA requería, además de contar con el parque Científico de Leganés Tecnológico, actualmente en construcción [7].

Entre las instituciones y departamentos de la Universidad que se encuentran vinculados al proyecto se encuentran las Humanidades, Periodismo y Comunicación Audiovisual, Documentación, Las Ciencias Jurídicas, la Escuela Politécnica Superior (Departamento de Informática, Teoría de la Señal y Tecnología Electrónica), el mencionado Parque Científico y Centro de Innovación Tecnológica, además del Instituto Pedro Juan de Lastanosa. Existe un serio compromiso por parte de la Universidad de conseguir resultados tangibles en el campo de la accesibilidad en televisión, subtítulo y audiodescripción, así como en la sensibilización social sobre la materia, en el periodo 2005-2006.

Entre las actividades más destacables de CESyA está la creación y mantenimiento de una base de datos de nombre SÁBADO (Subtitulado y Audiodescripción Base de Datos) constantemente actualizada donde recoger todo aquel material audiovisual subtítulo y/o audiodescrito en España. Se busca recoger de cada material subtítulo y/o audiodescrito sus características más generales y de interés para los usuarios del mismo, permitiendo encontrar de una manera cómoda y concreta los fondos subtítulos y/o audiodescritos que existen, quién posee los permisos de difusión, qué empresa lo subtitula o audiodescribe, etc., y agilizando todo el proceso de intercambio de ejemplares subtítulos o audiodescritos. La información será contrastada y verificada, creando un vínculo de confianza entre las distintas entidades que contribuyan con su material en la base de datos.

SÁBADO será un incentivo para la actividad del sector y no supondrá, en caso alguno, ningún tipo de competencia comercial con las empresas ya existentes. Todo lo contrario, su finalidad pasa por fomentar el intercambio y la circulación de material subtítulo y

audiodescrito entre las empresas productoras de dichos trabajos y los radiodifusores y demás clientes potenciales. Se constituye así como una herramienta de referencia en el mundo audiovisual que busca además homogeneizar y normalizar la producción de este tipo de material en España.

El acceso a la base de datos SÁBADO se realizará a través de una aplicación Web, tanto para las tareas de consulta y manipulación por parte de los usuarios autorizados a ello como para la realización de tareas de mantenimiento propias del usuario administrador. Mediante esta Web, un usuario previamente identificado, puede consultar toda clase de material audiovisual, insertar trabajos de subtítulo o de audiodescripción, modificar sus trabajos o materiales de los que es propietario, así como conocer los trabajos o datos de contacto de otras entidades vinculadas al CESyA. Igualmente, se trabaja en la posibilidad de sincronizar automáticamente mediante el intercambio de ficheros las diferentes bases de datos que las entidades vinculadas ya tienen con la base de datos del CESyA.

Es evidente, dada la naturaleza, propósitos y motivaciones del Centro, que es absolutamente imprescindible hablar de una Web plenamente accesible para discapacitados. Desde las primeras etapas de desarrollo, tanto la aplicación Web de acceso a la base de datos SÁBADO como la Web del CESyA han contado con un análisis, diseño e implementación orientados a conseguir el nivel de conformidad AAA. Todos los miembros del equipo sea cual fuere su función y su grado de responsabilidad (coordinadores, analistas, programadores, diseñadores Web, redactores, responsables de cualquier nivel) han trabajado con la idea siempre presente de conseguir dicha calificación.

Para ello, se han tenido que usar diferentes recursos y fuentes de información ya que, desgraciadamente hasta ahora, la accesibilidad en la Web no ha sido una característica frecuente. Toda la documentación sobre accesibilidad así como sus estándares existentes en el W3C [2] han sido una clara referencia durante el proceso de desarrollo de SÁBADO, pero no la única. Se hizo necesario buscar otros lugares de consulta en materia de accesibilidad, como foros y comunidades [30] que comienzan a aflorar en la red donde poder preguntar dudas sobre la materia, pedir consejos y aprender mutuamente técnicas para conseguir contenidos Web plenamente accesibles.

No sin poco esfuerzo y una constante revisión, se consiguió que los primeros prototipos de las Web tanto de CESyA [6] como de la base de datos SÁBADO llegaran, según las herramientas automáticas de validación de accesibilidad TAW [28] y HERA [29], al nivel de conformidad AAA.

Se puede comprobar así como CESyA, ya desde sus más tempranas etapas de existencia, fomenta la plena y total accesibilidad no sólo en aquel campo que constituye su principal y básica área de actuación, el sector audiovisual, sino también en los contenidos Web que dependen del Centro. Un principio realmente prometedor que augura tanto un éxito en la consecución de los objetivos de CESyA como también un éxito para los que durante tanto tiempo llevan reclamando que las nuevas tecnologías sean un arma contra la desigualdad y no una nueva barrera entre las personas.

3.4 Accesibilidad Web

A continuación se detalla una breve introducción a la accesibilidad en la Web, explicando los estándares existentes respecto a la materia, herramientas de validación y algunos recursos disponibles en Internet.

3.4.1 Pautas de Accesibilidad al Contenido en la Web

El W3C es una asociación internacional formada por organizaciones relacionadas con el mundo de Internet, personal y el público en general, que trabajan conjuntamente para desarrollar estándares Web. La misión del W3C es “*guiar la Web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la Web*” [26].

Para lograr su objetivo, el W3C publica sobre cada materia relacionada con el desarrollo Web un estándar o recomendación. Una recomendación oficial del W3C no es más que una especificación o serie de pautas que es aprobada por consenso por todos los miembros que forman parte del Consorcio. Algunas de estas recomendaciones cubren tecnologías como HTML, las hojas de estilo o CSS o el lenguaje XML, entre otras muchas.

En el campo de la accesibilidad Web, el W3C creó una entidad especializada denominada WAI [3]. El cometido de esta iniciativa es el de facilitar a las personas con alguna discapacidad el acceso a toda clase de contenidos Web, desarrollar pautas de accesibilidad, educar y concienciar de la importancia que tiene la accesibilidad a los desarrolladores Web y a la sociedad en general, mejorando las herramientas para construir páginas Web accesibles y fomentando la investigación en esta área.

Se dice que una Web es accesible cuando se posibilita un acceso universal a todos y cada uno de sus contenidos, independientemente del tipo de hardware o software que posee el usuario, su localización geográfica, su cultura, idioma o capacidades sensoriales. La accesibilidad implica que una persona con discapacidad pueda percibir, entender, navegar, interactuar e incluso contribuir a la Web, y esta accesibilidad se obtiene consiguiendo que dicha Web sea flexible para poder ser usada, sin perder su significado y contenido completos, bajo diferentes configuraciones, preferencias y dispositivos. De este modo, la accesibilidad no sólo está al servicio de la persona con alguna discapacidad, sino que también ayuda a aquellas personas que no la tienen, al estar accesibilidad y usabilidad íntimamente unidos [2].

Es importante hacer notar que, tal y como explica el W3C, la accesibilidad no debe verse únicamente como un intento de acercar las nuevas tecnologías a las personas discapacitadas, sino que una Web considerada verdaderamente accesible facilita igualmente el contenido de la misma tanto a personas discapacitadas como a las que no. Por ejemplo, bajo condiciones especiales de limitación sensorial o tecnológica, como serían personas de la tercera edad, ruido ambiental, dificultad en la conexión, navegadores limitados o anticuados, etc.

Así pues, para guiar en el diseño e implementación de páginas Web accesibles, el W3C ha desarrollado las denominadas “Pautas de Accesibilidad al Contenido en la Web” [27] (*Web Content Accessibility Guidelines*, WCAG son sus siglas en inglés). Este documento, publicado en 1999, tiene rango de recomendación por lo que constituye el documento final de un largo trabajo de los miembros del consorcio y marca el camino a seguir para cumplir con el estándar del W3C en cuanto a accesibilidad. Consta de un total de 14 pautas en cuanto al diseño Web y proporcionan soluciones a cada uno de los casos con los que se puede encontrar un desarrollador. Cada una de estas pautas constan

de una serie de puntos de verificación que comprueban hasta qué punto se cumple con la recomendación oficial de W3C. A su vez, estos puntos de verificación se dividen, según su importancia y alcance, en alguno de esto tres tipos:

- Punto de verificación de prioridad 1: aquél punto de verificación que un desarrollador Web tiene que cumplir ya que, de otra manera, ciertos grupos de usuarios **no podrían acceder** a la información del sitio Web.
- Punto de verificación de prioridad 2: aquél punto que un desarrollador Web debería cumplir ya que, si no fuese así, **sería muy difícil acceder** a la información para ciertos grupos de usuarios.
- Punto de verificación de prioridad 3: aquél punto de verificación que un desarrollador Web debería cumplir ya que, de otra forma, algunos usuarios experimentarían **ciertas dificultades para acceder** a la información.

Y en función de la prioridad de los puntos de verificación que una Web cubre, se puede entonces determinar qué grado de accesibilidad ofrece dicha Web:

- Nivel de conformidad A: todos los puntos de verificación de prioridad 1 se satisfacen.
- Nivel de conformidad AA: todos los puntos de verificación de prioridad 1 y 2 se satisfacen.
- Nivel de conformidad AAA: todos los puntos de verificación de prioridad 1,2 y 3 se satisfacen.

El citado documento, “Pautas de Accesibilidad al Contenido en la Web”, no es el único estándar del W3C en materia de accesibilidad, aunque sí el más importante y recurrido. Así, también se han desarrollado otras recomendaciones como las “Pautas de Accesibilidad para Herramientas de Autor” [33] (destinada a desarrolladores de herramientas software de autor para producir Web accesibles), las “Pautas de Accesibilidad para XML” [34], las “Pautas de Accesibilidad para Agentes de Usuario” [35] (en el que se detalla cómo hacer navegadores, reproductores multimedia y otro software accesible) y las prácticas “Técnicas para Pautas de Accesibilidad al Contenido en la Web”, que proporciona ejemplos y explicaciones pormenorizadas para trabajar con HTML y CSS accesible [36].



Figura 3-1: Símbolos de accesibilidad Web validada

Al igual que con otros estándares, el W3C ha creado un signo distintivo (mostrado en la Figura 3-1) para aquellas Web que han comprobado mediante las herramientas de validación correspondientes que cumplen con alguno de los niveles de conformidad anteriormente mencionados [2].

3.4.2 Herramientas de validación

Existe disponible en la red diverso software que permite comprobar de manera automática si una página Web cumple o no con las pautas marcadas desde el W3C [27]. Entre las principales herramientas de validación destacan por su amplio uso y eficiencia:

- **Test de Accesibilidad Web (TAW):** constituye una potente herramienta gratuita para el análisis de la accesibilidad de páginas Web, repasando minuciosamente todos los elementos que las componen. Esta aplicación está destinada tanto a profesionales (*webmasters*, desarrolladores, diseñadores, etc.) como a principiantes y público en general (estudiantes, *amateurs*, etc.) y se basa fielmente en las reglas marcadas desde el WAI [3]. Genera informes muy completos donde se indican cada uno de los puntos de verificación, su prioridad, la línea donde se encuentran y el número de ellos existente en la Web validada, entre otros muchos indicadores. Está disponible tanto para descargar e instalar en el propio equipo, como para instalar como barra de herramientas en navegadores Mozilla Firefox, así como para comprobar *online* la accesibilidad de una Web proporcionando su URL [28]. TAW está financiado conjuntamente por el Ministerio de Industria, Turismo y Comercio y por el Gobierno del Principado de Asturias, estando su desarrollo a cargo de la Fundación CTIC (Centro Tecnológico de la Información y la Comunicación, Principado de Asturias) [43].

- HERA: la herramienta de validación HERA permite revisar la accesibilidad de páginas Web siguiendo para ello las recomendaciones del W3C. HERA proporciona una validación automática y sugiere puntos de validación que deberían ser revisados manualmente para asegurar una completa comprobación accesible. El validador HERA permite indicar el tipo de comprobación que se desea realizar, añadir a la revisión el nombre de quién la ha llevado a cabo o incluir comentarios a los puntos de verificación. Así mismo, brinda la posibilidad de generar un informe final sobre la revisión, pudiendo ser un informe imprimible o descargable en varios formatos (XHTML, RDF, PDF). HERA es una aplicación gratuita y de libre uso, desarrollada enteramente en PHP y practica la filosofía del código abierto, pudiendo ser mejorada por aquel que lo desee. El proyecto HERA se llevó a cabo gracias a la Fundación SIDAR- Acceso Universal, que tiene como principal objetivo fomentar las acciones de investigación, formación, promoción, asesoría, etc., y todas aquellas actividades que faciliten el desarrollo de la Sociedad de la Información de forma accesible e inclusiva [29].

A pesar de la existencia de varios validadores automáticos no se debe olvidar además, como bien indican todos ellos tras mostrar sus resultados, de la importancia de realizar una comprobación manual sobre cada uno de los puntos de verificación. Esta revisión manual es necesaria debido a que existen algunos aspectos que las herramientas automáticas o bien no cubren o bien pueden darlas por válidas erróneamente sin tener en cuenta otros aspectos. La revisión manual será la que finalmente certificará si estamos o no ante una Web plenamente accesible.

3.4.3 Recursos Web sobre accesibilidad

La accesibilidad hasta hace relativamente poco era una asignatura pendiente en la mayor parte de los proyectos de desarrollo de una Web. Afortunadamente esta situación comienza a cambiar, y empiezan a surgir portales, foros, comunidades y páginas Web en general que tratan el tema de la accesibilidad, proporcionando material, consejos y toda clase de ayuda sobre la materia.

Discapnet.

Discapnet pasa por ser probablemente la Web más importante en español sobre discapacidad. Responde a una iniciativa para fomentar la inserción tanto social como laboral de aquellas personas con alguna discapacidad, y está financiada por la Fundación ONCE y el Fondo Europeo de Desarrollo Regional (FEDER). Como objetivos principales de este portal están proporcionar información de temas sobre discapacidad y el de proporcionar una plataforma para promover la participación en la sociedad de las personas discapacitadas. Entre sus servicios se encuentran noticias, fondos documentados sobre discapacidad, convocatorias de eventos, empleo *online*, asesoramiento personalizado, un callejero accesible y comunidad virtuales [30].

A pesar de ser un portal amplio y con una gran riqueza en cuanto a contenidos, Discapnet es una Web plenamente accesible. Facilita la navegación mediante una cabecera con ruta, cambios en la dimensión del texto, teclas de acceso rápido o la posibilidad de filtrar contenidos en función de los intereses del usuario, entre otros muchos aspectos. Discapnet dispone de documentos y ayudas sobre el desarrollo accesible de sitios Web, informa puntualmente sobre cualquier novedad en el mundo de la accesibilidad Web o audiovisual y recientemente ha constituido el Observatorio de la Infoaccesibilidad, destinado a realizar informes periódicos sobre la accesibilidad en las Web de las universidades españolas, las administraciones central y autonómicas y otros organismos de ámbito público.

Accesoweb.

Accesoweb es una popular lista sobre problemas y soluciones de diseño accesible para la red. Su objetivo es el de servir de medio de comunicación e información a los miembros del Seminario Iberoamericano sobre Discapacidad y Accesibilidad en la Red (SIDAR) [31]. Fue creada como un simple grupo dentro del portal Yahoo pero actualmente cuenta con cerca de 1200 usuarios registrados y cuenta en sus archivos con casi 10000 mensajes y consultas. Es un foro muy activo y su principal atractivo, además de registrar desde sus comienzos en 1999 cada una de las preguntas y respuestas realizadas, es la rapidez con la que los miembros más expertos responden a cualquier tipo de duda sobre accesibilidad Web.

Además del foro, cuenta con una sección de documentos de interés para la comunidad de usuarios como manuales sobre accesibilidad, concursos públicos, leyes, congresos y reuniones de profesionales y artículos de prensa o Internet.

3.5 Lenguajes de acceso a bases de datos

El lenguaje HTML ha sido hasta el momento y sin lugar a dudas la pieza más importante en la creación de páginas Web. HTML es un lenguaje de marcación de texto que sirve para estructurar y dar formato los documentos Web, convirtiéndolos en documentos de hipertexto entendible por los diferentes navegadores del mercado.

Con el paso del tiempo han ido surgiendo diferentes tecnologías y lenguajes que, si bien se siguen apoyando en este lenguaje como base y esqueleto fundamental de cualquier página, añaden funcionalidades y características que HTML no puede ofrecer. Así, hoy día, las páginas Web se pueden clasificar en dos grandes tipos:

- Páginas estáticas: se construyen en lenguaje HTML, no permite más efectos dentro de una página que los enlaces de hipertexto, imágenes o videos y sonidos.
- Páginas dinámicas: aquellas que incluyen cualquier efecto más elaborado o funcionalidad especial (menús desplegables, animaciones, botones que reaccionan con el puntero del ratón, capturas de eventos, etc.) y para ello es necesario utilizar otros lenguajes además de HTML.

A su vez, la páginas denominadas dinámicas se pueden dividir en dos subgrupos: las páginas dinámicas del lado del cliente (es el navegador del cliente el que procesa la página) y las páginas dinámicas del lado del servidor (el proceso de la página se lleva a cabo en el propio servidor donde están alojadas).

Las páginas del lado del cliente poseen la ventaja de ser rápidas y de ofrecer respuestas inmediatas a las acciones del usuario. Sin embargo, tienen un importante inconveniente y es que dependen mucho de las características del navegador donde son procesadas, lo que puede llevar a la situación de que lo que funcione en un navegador no lo haga en otro. Esto será perjudicial para aquellas personas que dispongan de un navegador

especial y diferente a los mayoritarios, como puede ser el caso de personas con alguna discapacidad.

Por otro lado, las páginas dinámicas del lado del servidor dejan toda la responsabilidad de ejecución y procesamiento al servidor desde el cual son ofrecidas (Figura 3-2). El inconveniente que esto representa es que el servidor ha de ser más potente, y además, dependiendo del caso, puede que pierda en escalabilidad y concurrencia con muchas solicitudes de servicio simultáneas. Aún así presentan una gran ventaja a tener muy en cuenta, y es que independientemente de las características del navegador que tenga el cliente, las páginas llegan a éste enteramente en lenguaje HTML, puesto que ha sido el servidor el encargado de ejecutarlas y transformarlas en algo legible para cualquier navegador, que al fin y al cabo, es siempre HTML. Algo verdaderamente importante para aquellos usuarios con navegadores no demasiado sofisticados o con limitadas funciones. Además, aspecto muy interesante de las páginas dinámicas, o mejor dicho de los lenguajes para hacer estas páginas, es que permiten, además de una bonita y vistosa presentación, conectar con muy diversas fuentes de bases de datos.

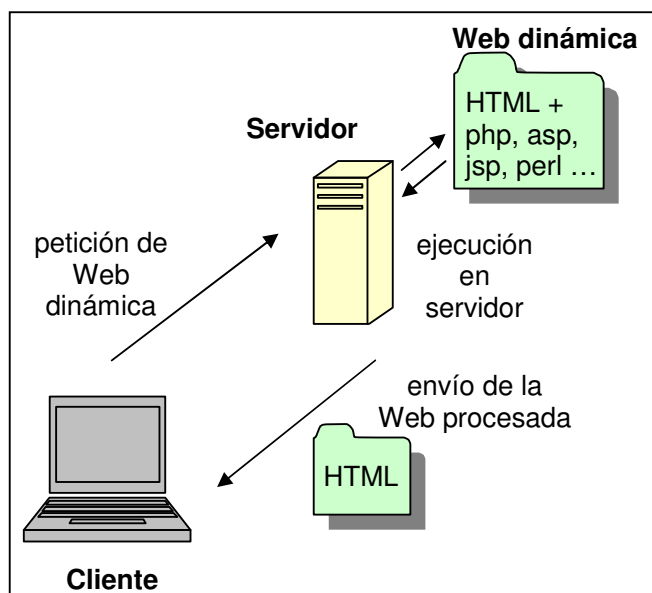


Figura 3-2 Funcionamiento de Web dinámica

3.5.1 ASP

ASP son las siglas de *Active Server Pages* y es una tecnología del lado del servidor desarrollado por la empresa Microsoft. Sirve para crear página Web dinámicas, embebiéndolas o no en el propio código HTML. La tecnología ASP, al ser desarrollada por Microsoft, solamente puede trabajar en servidores vinculados a Windows como IIS (*Internet Information Server*) o el PWS (*Personal Web Server*).

Generalmente ASP es utilizado sirviéndose del lenguaje Visual Basic Script que no es más una variante de Visual Basic. Aún así, también se puede utilizar Javascript para generar páginas en ASP. Únicamente se exige especificar con qué lenguaje se está usando. Hasta el momento, se han lanzado cuatro versiones principales, siendo la más reciente y última ASP .NET 2.0, disponible desde Noviembre de 2005.

Para aquellos equipos de desarrollo habituados al entorno Windows, ASP representa una muy buena opción ya que cuenta con una amplia documentación proporcionada por Microsoft y es una tecnología plenamente respaldada y certificada por dicha empresa. Además, permite interactuar con multitud aplicaciones Windows.

Las principales ventajas de ASP se consideran las siguientes:

- Gran comunidad de usuarios
- Amplia arquitectura de funcionamiento
- Complementa los entornos de desarrollo Web ya existentes de Microsoft
- Completa compatibilidad con entornos y aplicaciones Windows

Como desventajas, se puede mencionar que debe utilizarse conjuntamente con otros lenguajes como Visual Basic Script o Javascript (lo que implica aprenderlos si se desconocen), no es código abierto por lo que se necesita licencia, y aunque existen formas de trabajar con ASP fuera de Windows, su correcto funcionamiento no está plenamente garantizado.

3.5.2 JSP

JSP es un acrónimo de *Java Server Pages* y representa una tecnología destinada a crear páginas Web con programación en Java. Con JSP podemos crear aplicaciones Web que

se ejecuten en variados servidores Web de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma y JSP sigue la misma filosofía, conocida como "escribe una vez, ejecuta donde quieras".

El motor de las páginas JSP está basado en los *servlets* de Java, esto es, programas en Java destinados a ejecutarse en el servidor. El número de desarrolladores que pueden afrontar la programación de JSP es mucho mayor que el de Java, dado que resulta mucho más sencillo aprender que los *servlets*. Antes de que sean funcionales los archivos creados bajo este lenguaje, el motor JSP lleva a cabo una fase de traducción de esa página en un *servlet*, implementado en un archivo ".class" (*bytecodes* de Java).

Se deduce por tanto que para aprender JSP, aparte evidentemente de conocer HTML, será necesario comprender y tener algo de experiencia en la programación en Java, que es un lenguaje de programación orientado a objetos por completo. Una vez conocida la programación en Java se puede estudiar por encima el sistema de *servlets*, lo que nos dará una mejor idea del funcionamiento interno del motor JSP. Además, será necesario instalar Tomcat, el contenedor de *servlets* usado en la referencia oficial de implementación de JSP.

La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones Web, almacenando en clases Java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento HTML en el archivo JSP. Además Java se caracteriza por ser un lenguaje que puede ejecutarse en cualquier sistema, lo que sumado a JSP le da mucha versatilidad.

Sin embargo JSP no se puede considerar un *script* al 100% ya que antes de ejecutarse el servidor Web compila el *script* y genera un *servlet* y aunque este proceso sea transparente para el programador no deja de ser una aplicación compilada. La ventaja de esto es algo más de rapidez y disponer del API (siglas *Application Programming Interface*, Interfaz de Programación de Aplicaciones) de Java en su totalidad.

JSP presume de las siguientes ventajas frente a otros competidores:

- Es totalmente multiplataforma

- Relativamente fácil de aprender, si se conoce previamente el entorno Java
- Es rápido
- Integrado con las clases de Java.

Como inconvenientes, en caso de haber trabajado poco o nada con Java puede ser difícil de aprender sobre todo por su total adecuación a la orientación a objetos, además de necesitar la instalación de software adicional como Tomcat.

3.5.3 PHP

PHP es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación tanto en Internet en general como en su Web oficial [10]. Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones Web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos. También tiene la posibilidad de usar programación procedimental o programación orientada a objetos [10].

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X y RISC OS. Soporta la mayoría de servidores Web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Xitami y muchos otros. PHP tiene módulos disponibles para la mayoría de los servidores y para aquellos otros que soporten el estándar CGI (siglas de *Common Gateway Interface* o Pasarela de Interfaz Común), PHP puede usarse como procesador CGI.

Lo que más destaca la comunidad de usuarios de PHP es su facilidad de uso y aprendizaje para cualquier principiante mínimamente iniciado en la programación y a su vez representar una potentísima herramienta para los programadores y diseñadores avanzados. PHP puede hacer cualquier cosa que se pueda hacer con un *script* CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir *cookies*. El uso más extendido de PHP es la creación de

scripts del lado del servidor a través de páginas Web, pero también es útil para crear *scripts* en línea de comandos (por ejemplo, para el Planificador de Tareas de Windows o para tareas simples de procesamiento de textos) o incluso para desarrollar aplicación de interfaz gráfica. Con todo ello, las principales ventajas de PHP se podrían resumir como las siguientes:

- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Posee una muy buena documentación en su página oficial.
- Es un lenguaje multiplataforma.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).
- Permite las técnicas de programación orientada a objetos.

Pero sin duda la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos lo que permite escribir un interfaz vía Web para cualquiera de ellas utilizando PHP. Las principales bases de datos soportadas actualmente son Oracle, dBase, Interbase, PostgreSQL, Sybase, IBM DB2, MySQL o ODBC, entre otras [10].

Como desventajas, algunos desarrolladores encuentran PHP algo más lento que otros lenguajes en la llamada y ejecución de funciones [12].

3.5.4 Perl

Perl es un lenguaje de programación muy popular tanto en la construcción de páginas Web como en la de aplicaciones CGI. Es de libre uso, lo que implica que es gratuito. Hasta hace relativamente poco estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como por ejemplo Windows.

En sus orígenes, Perl tenía un cometido muy distinto al del desarrollo Web. Sus siglas hacen referencia a *Practical Extracting and Reporting Language*, que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros. Lo que en

Perl resulta absolutamente natural hacer es efectuar búsquedas de secuencias de caracteres dentro de cadenas, sustituciones de cadenas, operaciones sobre archivos de textos estructurados en campos o sin estructurar [14].

Además, Perl es extensible a partir de otros lenguajes, ya que podremos hacer llamadas a subprogramas escritos en otros lenguajes y también desde otros lenguajes podremos ejecutar código Perl.

Perl está inspirado a partir de lenguajes como C. Es por ello que un programador que haya trabajado con el lenguaje C y los otros tendrá menos problemas en entenderlo y utilizarlo rápidamente. Una diferencia fundamental de Perl con respecto a los otros lenguajes es que no limita el tamaño de los datos con los que trabaja, el límite lo pone la memoria que en ese momento se encuentre disponible.

Perl está recomendado para la realización de las siguientes funciones:

- Escritura de procesos CGI instalados en un servidor Web
- Desarrollo de procesos de mantenimiento de las actividades de un servidor

Sin embargo, Perl está desaconsejado para procesos de puro cálculo científico o programas que precisen de una gran velocidad y precisión en cálculo, así como elaboraciones numéricas complejas [14].

3.5.5 Elección de un lenguaje de acceso a bases de datos

Una vez estudiados las diferentes alternativas existentes en la programación de páginas dinámicas con acceso a bases de datos se debe escoger uno de ellos para la realización del proyecto. ASP y JSP aparecen como opciones muy potentes y extendidas, pero implican el aprendizaje o conocimiento de otros lenguajes en los que se basan (Visual Basic y Java, respectivamente). Además, en el caso de JSP, implica también la instalación de Tomcat con los *servlets* de Java.

Perl se perfila igualmente como una buena opción, pero sus mayores usos se dan bajo entornos Unix a pesar de existir cada día más usuario en sistemas Windows. PHP surge

como la mejor opción entre todos ellos, debido a su versatilidad, su fácil aprendizaje, su amplísima documentación oficial y gran comunidad de usuarios, así como el alto número de SGBD que soporta.

3.6 Sistemas Gestores de Bases de Datos (SGBD)

Los SGBD son programas dedicados exclusivamente a ofrecer una interfaz entre las bases de datos y las aplicaciones que las usan a través del sistema operativo. Tradicionalmente, los SGBD se componen de un lenguaje de definición de datos, un lenguaje de manipulación de datos, y un lenguaje de consulta.

Utilizando un SGBD se consigue un acceso a los datos más fácil de implementar, más seguro y sobre todo más eficiente. Además, los SGBD pueden proporcionar servicios adicionales como control de la redundancia, limitación de accesos, independencia entre datos y aplicación, o mantenimiento de la integridad de los datos, entre otros.

3.6.1 Microsoft SQL Server

Microsoft SQL Server (o simplemente SQL Server) es el gestor de bases de datos relacionales de Microsoft basado en el lenguaje SQL y constituye uno de los SGBD más utilizados en el mundo. En realidad, SQL Server utiliza una variante del estándar SQL-92 llamada T-SQL o Transact-SQL. A diferencia de otros, no es multiplataforma puesto que solamente puede trabajar bajo sistemas operativos de la familia Windows.

A pesar de ser un SGBD comercial, existe una versión limitada pero con muy buenos resultados hasta ahora con bases de datos pequeñas o medianas llamada SQL Server Express.

Entre sus características más destacables podemos encontrar:

- Transacciones
- Escalabilidad
- Procedimientos almacenados
- Estabilidad y seguridad
- Posibilidad de trabajar en arquitectura cliente-servidor

Su última versión es Microsoft SQL Server 2005. Entre sus mejoras destaca una mayor disponibilidad, mayor eficiencia en la elaboración de copias de seguridad y su posterior recuperación, partición de tablas e índices reduciendo el tiempo en las consultas, mayor seguridad, soporte para XML, nuevas herramientas gráficas de gestión e interoperabilidad con otras aplicaciones para Windows.

Una importante opción que presenta Microsoft SQL Server es la posibilidad de trabajar conjuntamente con Microsoft Access, un sencillísimo y limitado gestor de bases de datos, pero muy común y ampliamente extendido en todo tipo de empresas. De esta forma se completa una potente base de datos como es SQL Server con un entorno de desarrollo cómodo y de alto rendimiento como VBA Access a través de la implementación de aplicaciones mediante el uso de formularios Windows (la gama VBA permite a usuarios y programadores ampliar la funcionalidad de programas Microsoft). Para el desarrollo de aplicaciones más complejas, SQL Server incluye interfaces de acceso para la mayoría de las plataformas de desarrollo, incluyendo la popular .NET.

El principal inconveniente de SQL Server es, incluyendo el tener que pagar por su uso, la imposibilidad de trabajar con él fuera de entornos Windows. Además, a pesar de que sus creadores presumen de que la última versión lo convierten en uno de los SGBD más seguros, una vulnerabilidad en su código no detectada a tiempo provocó que el gusano conocido como *SQL slammer* causara en enero de 2003 a lo largo de todo el mundo una de las mayores caídas del sistema conocidas hasta ahora en Internet [11].

3.6.2 Oracle

Oracle es un software gestor de bases de datos fabricado por Oracle Corporation y constituye probablemente el gestor más popular y vendido del mundo. La última versión lanzada fue en 2006 Oracle Database 10g. Se le considera uno de los SGBD más completos destacando:

- Estabilidad
- Soporte de transacciones
- Funciones y procedimientos almacenados

- Escalabilidad
- Multiplataforma (las últimas versiones han sido plenamente certificadas para poder trabajar en Linux)

Hasta ahora, su dominio en el mercado ha sido casi absoluto, aunque en los últimos años le han surgido importantes competidores como Microsoft SQL Server o incluso software gratuito como MySQL o PostgreSQL.

El principal defecto de Oracle es su elevado precio que puede llegar a varios miles de euros dependiendo de la versión adquirida y la licencia, lo que hace que sólo se vea por norma general en empresas muy grandes o multinacionales [16]. Algunos expertos también critican los importantes fallos de seguridad que en ocasiones ha demostrado tener.

3.6.3 Firebird

El sistema gestor de bases de datos Firebird es software libre o de código abierto cuyo desarrollo está basado en la versión 6 de Interbase, otro SGBD muy potente y de pago. El código fue liberado en 1999 por Borland (que compró la compañía desarrolladora de Interbase) y reescrito completamente en C++, corrigiendo muchos de los errores que tenía. En este momento, Interbase y Firebird son prácticamente iguales, aunque seguirán en el futuro caminos completamente diferentes, manteniendo el uno su licencia de pago y el otro su código abierto.

A pesar de ser de reciente aparición y de competir directamente con los más grandes y extendidos SGBD, entre las principales características de Firebird destacan:

- *Triggers* o disparadores
- Procedimientos almacenados
- Integridad referencial
- Multiplataforma
- Acceso basado en usuarios y roles
- Arquitectura cliente-servidor
- Lenguaje SQL compatible con SQL-92
- Funciones definidas por el usuario

- *Driver* nativo para PHP

El principal inconveniente de Firebird es que hoy en día está muy poco utilizado, siendo difícil encontrar una comunidad de usuarios amplia o documentación adicional a la ofrecida en su página Web.

3.6.4 MySQL

MySQL es un SGBD desarrollado bajo la filosofía de código abierto y se ha convertido en uno de los sistemas de bases de datos más populares en todo el mundo en esta categoría. Aunque pueda utilizarse gratuitamente y su código fuente esté disponible, también existe una versión de pago para aquellos que quieran darle un uso con fines lucrativos.

A diferencia de otros proyectos de código abierto en los que los usuarios amplían y mejoran las prestaciones del producto y nadie es propietario de nada, la compañía MySQL AB es la propietaria de este SGBD, siendo la encargada de desarrollarlo, mantenerlo y ampliarlo, al igual que de vender las licencias en los casos en los que ésta es necesaria. Aún así, el código siempre está disponible para todo el que desee conocerlo.

Según las cifras del fabricante, actualmente existen en el mundo más de 5 millones de copias instaladas y una comunidad de usuarios que supera en número a la de cualquier otro SGBD. Diariamente, 35000 nuevos internautas se descargan la última versión estable (MySQL 5.0, lanzada a finales de 2005) desde alguno de sus servidores oficiales [18].

Entre las características disponibles en las últimas versiones se puede destacar:

- Disparadores
- Procedimientos almacenados
- Disponibilidad en gran cantidad de plataformas y sistemas
- Transacciones
- Claves foráneas
- Replicación

- Amplio subconjunto del lenguaje SQL
- Diferentes configuraciones de almacenamiento
- Conectividad segura
- Búsqueda e indexación de campos de texto

MySQL es muy utilizado para aplicaciones Web debido probablemente a la también popularidad de PHP, con el que a menudo es utilizado. Hay una extensa documentación que hace referencia al uso de los dos en la construcción y desarrollo de páginas Web con acceso a datos. Se suele afirmar por parte de su comunidad de usuarios que MySQL es más fácil de aprender y de usar que la mayoría de los SGBD.

Entre las críticas recibidas se encuentra que carecía hasta hace poco de algunas funcionalidades importantes sí implementadas en otros SGBD, como por ejemplo transacciones o integridad referencial. Estas objeciones han quedado en evidencia tras la última versión estable de MySQL. Otros detractores lamentan que MySQL sea menos respetuoso con el estándar de SQL que sus competidores.

3.6.5 PostgreSQL

Postgre SQL es un sistema gestor de bases de datos relacionales de código abierto y pretende ser una alternativa a otros SGBD tanto libres como propietarios. Detrás de PostgreSQL se encuentran muchos años de investigación y desarrollo de varios prototipos y versiones limitadas, hasta que finalmente el proyecto fue abandonado en 1994. La licencia bajo la cual PostgreSQL había sido liberado permitió a usuarios anónimos y desarrolladores de código abierto obtener una copia de su código fuente para continuar desde entonces con su desarrollo.

Algunas de sus principales características son:

- Claves ajenas
- Disparadores
- Vistas
- Integridad referencial
- Multiplataforma
- Acceso concurrente

- Capacidad de albergar programas en el servidor en varios lenguajes.
- Herencia de tablas
- Tipos de datos y operaciones geométricas

Según numerosas pruebas, el equipo desarrollador de PostgreSQL asegura que éste SGBD es más fiable y estable que cualquier otro del mercado, ya sea de código abierto o comercial [19]. Como parte negativa, PostgreSQL adolece de una menor aunque creciente comunidad de usuarios y quizás ciertas dificultades para su instalación completa.

3.6.6 SQLite

SQLite es un completo motor de base de datos que no precisa configuración y constituye una pequeña librería programada en lenguaje C. SQLite destaca, además de por su velocidad, por su versatilidad. En la última versión estable de PHP 5 se incluía por fin librerías y funciones para SQLite.

A diferencia del paradigma usual cliente-servidor, el motor de SQLite no es un proceso autónomo con el que el programa se comunica, sino que está unido y es por eso que se convierte en parte integral del programa. El protocolo principal de comunicación hace llamadas directas a la API desde el lenguaje de programación. La base de datos completa (definiciones, tablas, índices, y los propios datos) queda almacenada en un simple fichero estándar en el servidor.

La librería implementa la mayoría del estándar SQL-92, incluyendo transacciones, que son ACID (*Atomicity, Consistency, Isolation, Durability* o en español indivisibilidad, consistencia, aislamiento y durabilidad), disparadores y la mayoría de consultas complejas. No hay comprobación de tipos: se puede insertar una cadena en una columna de enteros, por poner un ejemplo. Algunos usuarios lo ven como una innovación muy útil, especialmente cuando está unido a lenguajes de tipos dinámicos. Otros usuarios lo ven como una gran y peligrosa pega.

Es muy rápido y la ventaja fundamental es que permite utilizar el lenguaje estándar SQL. Como contrapartida, parece destinada principalmente a aquellos profesionales que

estén muy familiarizados con el mundo de la programación y el código abierto, no siendo apto para usuarios con poca preparación.

3.6.7 Elección de un SGBD

Tratándose de un proyecto destinado a una ONG, parece obvio que el primer criterio que se debe establecer es el económico. Dicho esto, los SGBD por los que hay que pagar una licencia, SQL Server de Microsoft y Oracle, no se perfilan como una buena elección.

Dentro del software gratuito, Firebird es de reciente aparición y no cuenta con una amplia comunidad de usuarios, aunque ésta sí sea creciente. SQLite parece demasiado complejo de instalar y mantener, debido a que más que un programa en sí es una librería de funciones implementada en C. PostgreSQL y MySQL, gracias a su gran fiabilidad y mayor número de usuarios, surgen como las dos alternativas posibles. Finalmente, será MySQL el SGBD elegido debido a su mayor comunidad de usuarios, la existencia de múltiples ejemplos y documentación en Internet, así como los excelentes resultados obtenidos por su uso conjunto con PHP.

3.7 Servidores Web

Servidor Web es un término un tanto ambiguo hoy en día, puesto que con la palabra servidor se identifica tanto un software determinado que ofrece un servicio concreto como la máquina física donde este está instalado. Un Servidor Web, entendiéndolo como aplicación, es un programa que implementa el protocolo HTTP, utilizado para transferir páginas Web y todos sus elementos como textos con enlaces, figuras, imágenes, formularios, botones y demás objetos incrustados. El servidor se mantiene siempre “escuchando” posibles peticiones de servicio por parte de lo que se denomina cliente Web o navegador.

El proceso siempre funciona de la misma forma: el cliente o navegador realiza una petición al servidor que aloja la página Web que desea visionar, el servidor recibe la petición, procesa la página alojada en él en caso de ser una Web dinámica, la envía

mediante protocolo HTTP y el cliente recibe código HTML que interpreta mostrando el contenido de la Web con la estructura y el formato especificados.

3.7.1 Apache

El servidor Apache es un servidor HTTP de código abierto, gratuito y multiplataforma. Este servidor se desarrolla dentro del proyecto *HTTP Server* de la organización *Apache Software Foundation*, fundación que persigue los siguientes objetivos: el desarrollo y mantenimiento de software de código abierto, recogida de fondos de empresas o particulares para financiar sus investigaciones y coordinación de la amplia comunidad de usuarios y desarrolladores de sus productos de software libre [21].

Apache se puede considerar como el servidor Web más utilizado en todo el mundo, con aproximadamente un 70% del mercado [22].

Este servidor Web presenta una arquitectura fuertemente modular. La aplicación consta de un núcleo principal apoyado por una larga lista de funcionalidades que se implementan a través de módulos. Muchos de estos módulos pueden ser fácilmente ampliados para aumentar las prestaciones ya ofrecidas por defecto como, por ejemplo, incluir el procesamiento de páginas dinámicas del lado del servidor.

Además de la amplísima documentación existente por toda la red y la incondicional comunidad de usuarios de Apache, este servidor Web ofrece como ventajas mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Uno de sus puntos débiles es no contar con un entorno gráfico que facilite su tan variada configuración, ya que actualmente debe hacerse mediante la edición directa de un archivo de texto.

3.7.2 IIS

IIS, *Internet Information Services* (o en ocasiones *Internet Information Server*), es una serie de servicios para aquellos equipos que funcionan con Windows. Se le considera el servidor Web más utilizado del mundo después de Apache. Los servicios que ofrece esta aplicación son:

- FTP (protocolo de transmisión de archivos)
- SMTP (correo electrónico)
- NNTP (grupos de noticias)
- HTTP (transmisión de hipertexto).

Este programa convierte cualquier ordenador en un servidor de Internet o una intranet, pudiendo publicar páginas Web tanto de forma local como remotamente. El servidor Web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas como, por ejemplo, los incluidos para ASP y ASP.NET. Esto no significa que no puedan ser añadidos los de otros fabricantes, como PHP o Perl.

La última versión lanzada fue IIS 6.0 para Windows Server 2003, aunque se espera que con el lanzamiento del próximo sistema operativo Windows Vista aparezca también la versión IIS 7.0, con considerables mejoras en seguridad y modularidad.

IIS representa un software muy potente puesto que además de ofrecer páginas Web es capaz de realizar otro tipo de servicios como correo electrónico o transferencia de archivos. Sin embargo, además de no ser gratuito y funcionar con fiabilidad únicamente bajo sistemas Windows, la versión incorporada por defecto a Windows XP (IIS 5.0) está considerablemente limitada con respecto a la versión más reciente.

3.7.3 Cherokee

Cherokee es un proyecto que nació por el interés de una pequeña comunidad de usuarios de software libre por implementar un servidor Web que mejorase alguna de las carencias de los servidores que existen. La idea inicial del proyecto fue de un joven ingeniero español afincado en Irlanda, pero pronto muchos se sumaron a la creación de este nuevo servidor [25]. Se buscaba desarrollar un servidor Web con una arquitectura y funcionamiento más eficiente, pero que al mismo tiempo con flexibilidad suficiente como para poder ser sucesivamente ampliado y mejorado con cualquier funcionalidad que hoy en día pueda ser deseable en aplicaciones de este tipo.

Hasta ahora los servidores Web se basaban en procesos, hilos, en *sockets* no bloqueantes y los ambiguos servidores basados en la ejecución en el núcleo o *kernel*. La nueva filosofía aplicada por Cherokee es, a grandes rasgos, la de una aplicación que procesa varias peticiones de servicio en cada uno de sus hilos. Estos hilos ni son creados

ni destruidos, sencillamente se generan una vez que arranca el servidor y permanecerán vivos hasta que éste termine su ejecución [23].

Actualmente existen versiones instalables de Cherokee [24], todas ellas en sistemas Unix, y aunque sus pruebas de rendimiento, eficiencia, velocidad y facilidad de uso son muy prometedoras, todavía está lejos de ser un servidor Web de gran implantación.

3.7.4 Elección de un servidor Web

Una vez más, el criterio económico implica desechar el servidor de pago IIS de Microsoft. Por otro lado, a pesar de su uso gratuito y los excelentes resultados obtenidos por el servidor Cherokee, su reciente aparición y su única disponibilidad bajo entornos Unix obliga igualmente a no considerarlo como una posible opción.

Así pues, debido a su abrumador dominio del mercado, sus numerosa comunidad de usuarios, la amplia documentación disponible en Internet y la gran fiabilidad mostrada hasta ahora, Apache se perfila como el servidor Web más aconsejable para la realización de este proyecto.

4 TOMA DE REQUISITOS Y ANÁLISIS

4.1 *Contacto con FEDACE*

Como se ha explicado en el capítulo anterior, mediante la Web de voluntariado Haces Falta [4] se conoció la existencia de FEDACE y su petición de personal voluntario, preferiblemente con estudios o conocimientos en el desarrollo de bases de datos, para la realización de un proyecto informático en la ONG. El único trámite consistió en rellenar un formulario de candidato con el perfil académico, experiencia laboral y disponibilidad horaria.

La comunicación entre el cliente y el autor del presente proyecto se produjo a través de una llamada telefónica, donde se hizo una breve entrevista para confirmar los datos enviados en el formulario, conocer más a fondo el perfil académico y profesional del candidato y hacer un breve resumen del problema. Para finalizar, se concertó una entrevista para la definición detallada del proyecto y la toma de requisitos consecuente.

4.2 *Preparativos a la reunión*

Para hacer una toma de requisitos completa y aprovechar al máximo la reunión mantenida con el cliente, se decidió redactar una serie de cuestionarios¹ que facilitasen la obtención de los objetivos principales y complementasen la explicación del problema dada desde el cliente.

El cuestionario destinado a la base de datos se redactó utilizando para ello como modelo orientativo otros cuestionarios utilizados en la beca que se desempeñó en CESyA². Allí también se pasó, lógicamente, por una etapa de toma de requisitos y se utilizaron igualmente cuestionarios para estandarizar las preguntas y respuestas de las diferentes partes que hacían de clientes. Las preguntas no podían dejar de ser muy generales y poco específicas, puesto que todavía no se conocían los pormenores de la base de datos ni las características deseadas. El cuestionario únicamente pretendía servir

¹ En los anexos se encuentran disponibles los cuestionarios utilizados

² Para más información, consultar el capítulo 3, *Estado del Arte*

de guión durante la entrevista para obtener una idea general del problema planteado. De esta forma, las cuestiones abordaban temas como el por qué de la necesidad de una base de datos, cuál era el sistema seguido hasta ahora, quién trabajará con la base de datos, o cuestiones relacionadas con las prestaciones de la misma (velocidad, disponibilidad, número estimado de operaciones, etc.).

Con respecto a la aplicación Web, el cuestionario que se redactó se basó en la experiencia propia del autor del proyecto en el desarrollo de páginas Web y en algunas páginas en Internet destinadas a autodidactas o semiprofesionales en la creación de entornos Web [9]. Se hacían preguntas tanto relacionadas con los hábitos del cliente (si está familiarizado o no con el uso de Internet, tipo de explorador utilizado, motivos de la conexión, etc.) como preguntas más vinculadas a la estructura de la propia página (ampliación de la Web existente de FEDACE [5] o creación de una nueva, disponibilidad o no de plantillas u hojas de estilo, preferencias en cuanto a colores o estilo, elementos corporativos, tipos de letra, estilo de los formularios, etc.). Para este último tipo de cuestiones se optó por la creación de un simple y básico prototipo de ejemplo donde el cliente pudiera comprobar las alternativas planteadas en el cuestionario con respecto al aspecto y estructura de la ampliación Web.

4.3 Exposición del problema

FEDACE es una federación que agrupa a más de una veintena de asociaciones de afectados y familiares de afectados por daño cerebral en España. Su cometido principal es el de proporcionar información y orientar a los afectados, coordinar a las diferentes asociaciones y servir de interlocutor en las gestiones frente a la Administración. Entre los servicios de los que dispone se encuentra una amplia y exhaustiva recopilación de toda clase de recursos sobre daño cerebral, como centros de atención, hospitales, centros de rehabilitación, residencias, centros de orientación laboral y académica, centros de día, otras asociaciones, etc., con todos sus datos de contacto, servicios ofrecidos y equipo médico, entre otros.

Hasta ahora, de forma más o menos periódica, FEDACE redactaba toda esta información en una guía de recursos escrita que enviaba a las diferentes asociaciones

federadas o que ofrecía a las personas que acuden a su oficina en busca de ayuda y orientación. Cuando se creó esta iniciativa, en las primeras versiones de la guía se recogieron datos únicamente de los hospitales públicos y de las asociaciones de las más importantes ciudades españolas, pero actualmente esta guía ha aumentado considerablemente. En la última edición, la guía cubría prácticamente todo el territorio nacional y contenía información sobre cerca de 200 recursos de ámbito público o privado además de las distintas asociaciones creadas entorno al daño cerebral [Guía de recursos de atención al daño cerebral, 2005].

El proceso de recogida de datos sobre los recursos suele ser bastante lento y laborioso. En algunos casos FEDACE era la que tomaba la iniciativa y llamaba uno a uno a los centros ya conocidos para recabar aquellos datos de interés para la guía, mientras que en otros casos eran las propias asociaciones las que proporcionaban un informe con una serie de datos estándar, fijados desde FEDACE, de aquellos recursos de los que tenían conocimiento en sus ciudades o ámbito geográfico de actuación, ya que las asociaciones existen tanto a nivel local como autonómico.

Ante la importancia y dimensiones que comenzaba a adquirir esta guía de recursos, FEDACE deseaba disponer de una base de datos automatizada para poder ir almacenando de forma más cómoda toda la información sobre los distintos recursos, a la vez que facilitaría la consulta de los mismos, adaptándola a las necesidades concretas y específicas del interesado. Las distintas clases de recursos existentes, así como los servicios ofrecidos por multitud de disciplinas médicas distintas, hacen que los tipos de datos recogidos en cada uno de ellos sean poco homogéneos entre sí. Esta circunstancia, unida a la gran variedad de dolencias que puede haber entre personas afectadas por daño cerebral, dificulta la búsqueda del tipo de centro más apropiado para cada afectado si no se dispone de un método automático donde indicar qué es lo que se está buscando y para quién.

Además de todo ello, FEDACE exploraba la posibilidad de unir de alguna forma esta futura base de datos a la Web ya existente de la federación, que fue creada y es administrada actualmente por una empresa externa. El abanico de posibilidades de esta unión les parecía algo prometedor, pero necesitaban el asesoramiento de una persona con conocimientos en el área para poder dilucidar qué es lo que se podía hacer. En

concreto, la idea que FEDACE tenía de esta conexión entre la Web y la futura base de datos es la de crear, mediante un sistema de usuarios y acceso restringido, no sólo una aplicación desde la cual se pudiera acceder a consultas de todo tipo sobre los recursos que se fuesen añadiendo, sino también una aplicación sencilla y rápida para poder insertar o editar cualquier elemento dentro de la base de datos. Además, se barajaba la posibilidad de hacer aún más participativos a los futuros usuarios permitiéndoles la inserción de comentarios narrando la experiencia personal que han tenido con cada uno de los recursos, de forma que esta información sea también visible para el resto de usuarios que consulten esos recursos concretos.

FEDACE deseaba también que la futura aplicación Web contemplase los medios necesarios para ser completamente accesible, ya que era muy probable que algunos de los posibles usuarios fueran especialmente sensibles con esta circunstancia, además de que la propia naturaleza de FEDACE como federación de asociaciones de personas con daño cerebral exige obviamente prestar atención a los usuarios que presenten alguna discapacidad.

Para la acotación y concreción del problema, el personal de FEDACE rellenó los cuestionarios preparados para la reunión acerca de la base de datos y de la aplicación Web, además de exponer el tipo de diseño que deseaban en ésta última a través de la plantilla de prueba mostrada. Además, se proporcionó al autor del proyecto la guía de recursos recientemente editada para poder estudiar y analizar con mayor exactitud los datos que se pretendían almacenar en la base de datos [Guía de recursos de atención al daño cerebral, 2005]. Por último, se tomó nota de las características técnicas del equipamiento informático de las oficinas de FEDACE a fin de asegurar una elección correcta del software y herramientas idóneas para la materialización de los objetivos.

4.4 Análisis de los requisitos de usuario

Una vez mantenida la reunión con FEDACE, expuesto el problema y rellenados los cuestionarios para la toma de requisitos, es el momento de analizar toda la información recogida para obtener un perfil de cuál es el alcance del sistema y qué estrategias se deben seguir para la consecución de objetivos.

4.4.1 Análisis de la base de datos

El análisis de la base de datos comprende aspectos como el tipo de datos que se van a tratar o el SGBD a utilizar para la implementación física. Antes de ello, se exponen los requisitos de usuario obtenidos sobre los cuales se va a realizar el análisis posterior.

4.4.1.1 Requisitos específicos de usuario sobre la base de datos

Mantenida la reunión con FEDACE y rellenados los cuestionarios, el desarrollo de la base de datos puede llevarse a cabo siguiendo los siguientes requisitos:

- La base de datos debe reflejar los recursos existentes en la guía de recursos editada por FEDACE.
- Estos recursos deben tener los atributos y campos que les correspondan.
- Se desea ampliar la información ya existente añadiendo comentarios o evaluaciones personales de los usuarios de los recursos. Estos comentarios serán insertados por los propios usuarios.
- La base de datos se instalará en las oficinas de FEDACE y será administrada desde la misma.
- Únicamente puede insertar y editar datos de los recursos el administrador de la base de datos, esto es, FEDACE, con la excepción de los comentarios.
- Los otros usuarios podrán simplemente consultar recursos y escribir comentarios sobre los mismos.
- El desarrollo de la base de datos tendrá el mínimo coste asociado.

4.4.1.2 Análisis de los datos

La guía de recursos facilitada por FEDACE [Guía de recursos de atención al daño cerebral, 2005] correspondiente a la última edición, sirvió como referente a la hora de analizar el tipo de información que se iba a almacenar en la futura base de datos. Así, se pudieron observar, entre los casi 200 recursos existentes, las siguientes características:

- Los recursos pueden ser divididos en dos grandes bloques: las asociaciones y los centros de atención.

- Existen algunos recursos que se pueden englobar dentro de los dos tipos al mismo tiempo, esto es, asociaciones que además poseen instalaciones propias donde ofrecer alguna de las prestaciones típicas de los centros de atención.
- Dentro de las asociaciones, la gran mayoría corresponden a asociaciones de afectados mientras que una pequeña parte lo son de profesionales de algún área médica concreta.
- Los centros presentan una gran variedad en cuanto a su finalidad: centros de día, de atención domiciliaria, hospitales, residencias, centro de formación o inserción laboral y centros de rehabilitación, entre otros muchos.
- No existen atributos especiales para cada tipo de centro.
- A su vez, todos los centros de atención también se dividen en centros de ámbito público (ya sea estatal, autonómico o local) o de ámbito privado.
- La guía está estructurada siguiendo un criterio geográfico, es decir, agrupa los recursos por comunidades autónomas y sus provincias.
- Existen también, aunque pocos, asociaciones y centros de ámbito nacional, no circunscritos a ninguna comunidad autónoma concreta.
- En función de ser una asociación o un centro, los recursos difieren en el número y tipo de campos.
- Las asociaciones tienen un campo dedicado a la presidencia, gerencia o administrador principal de la misma.
- Los centros contienen una serie de informaciones no compartidas con los recursos que son asociaciones. Los principales atributos son:
 - Perfil o diagnóstico médico de los afectados atendidos en el centro
 - Forma de acceso al centro o criterios de admisión
 - Si son de titularidad pública o privada
 - Modelo de atención que ofertan (domiciliaria, de tipo hospitalario, rehabilitador, formativo, etc.)
- Los recursos cuentan también con una serie de servicios que ofrecen a los afectados por el daño cerebral. Estos servicios son de toda índole y naturaleza, pasando por una gran variedad de tratamiento médicos, servicios de rehabilitación, ocio y tiempo libre, terapias de grupo para familiares y afectados, programas de prevención y sensibilización social, etc.
- En un importante número de recursos, aparece un listado del tipo y número de personal con el que cuentan en sus instalaciones, ya sean médicos de alguna

especialidad, profesores, psicólogos, animadores socio-culturales, trabajadores sociales, administrativos u otro tipo de profesionales.

- La gran mayoría de los recursos, además de la información puntual acerca de sus datos de contacto y actividades, cuentan una descripción detallada o resumen donde se explican mejor y pormenorizadamente cada uno de los tratamientos aplicados, servicios ofrecidos, objetivos y naturaleza del recurso.
- Muchos de los recursos, ya sean asociaciones o centros, presentan información de contacto con alguna de las personas que trabajan en ellos, ya sean personal médico, administrativo o de atención al público.
- Algunos recursos estaban presentes en diferentes provincias, ofertando en cada una servicios específicos y disponiendo de un personal médico distinto que en otras.
- Casi todos los recursos tienen información acerca de su dirección, números de teléfono, de fax, dirección de correo electrónico, dirección de su página en Internet y otros datos de interés.

Cabe destacar la diferencia en cuanto a datos recogidos entre unos recursos y otros. En algunos de ellos, todos y cada uno de los campos anteriormente explicados aparecen, mientras que en otros simplemente reflejan el nombre del recurso, su dirección y quizás algún teléfono de contacto.

Por otro lado, algunos de los datos se han analizado más en profundidad para saber cuáles son los valores que frecuentan tener. Así, se consiguió obtener un listado para los siguientes campos:

1. Tipo de recursos existentes:
 - Asociación de afectados
 - Asociación de profesionales
 - Centro de atención
2. Dependencia o titularidad de los centros de atención:
 - Público
 - Privado concertado
 - Privado lucrativo

- Privado no lucrativo

3. Modelo de atención o clasificación de los centros.

- | | |
|--------------------------------------|--|
| - Centro de Día. | - Centro Residencial |
| - Centro de orientación laboral | - Hospital |
| - Centro de orientación pedagógica | - Hospital media-larga estancia |
| - Centro de Respiro | - Mini Residencias |
| - Centro Especial de Empleo | - Piso Tutelado |
| - Centro Ocupacional | - Servicio de Rehabilitación Domiciliaria. |
| - Centro Rehabilitación Ambulatoria | - Unidad de Estado Vegetativo |
| - Centro Rehabilitación Hospitalaria | |

4. Servicios ofrecidos por todos los recursos en general.

- | | |
|---------------------------------------|--|
| - Actividades de ocio y tiempo libre. | - Información, orientación y asesoramiento a las familias. |
| - Apoyo hospitalario | - Logopedia. |
| - Asesoría jurídica. | - Medicina general. |
| - Atención psicológica. | - Musicoterapia. |
| - Ayuda a domicilio. | - Neurocirugía. |
| - Grupos de ayuda mutua. | - Neurofisiología. |
| - Charlas. | - Neurología. |
| - Comedor. | - Neuropsicología. |
| - Cursos de formación a profesionales | - Neuropsiquiatría. |
| - Enfermería. | - Orientación e inserción académica. |
| - Evaluación neuropsicológica | - Orientación e inserción laboral. |
| - Fisioterapia. | - Ortopedias. |
| - Formación a la familia | - Piscina. |
| - Habilidades sociales y autoestima. | - Prevención y difusión del daño cerebral. |
| - Información y divulgación. | - Psicomotricidad |

- Psicoterapia.
- Psiquiatría.
- Revista.
- Radiología neuronal.
- Rehabilitación neuropsicológica
- Sensibilización social.
- Servicios generales (englobaría servicios tales como cocina,
- limpieza, lavandería,
- mantenimiento y cafetería).
- Talleres.
- Terapia de grupo.
- Terapia familiar.
- Terapia ocupacional.
- Trabajo social.
- Transporte.
- Voluntariado.

5. Tipo de personal profesional con el que cuentan los recursos en general

- Administrativo.
- Animador socio-cultural
- Audiometrista.
- Auxiliar administrativo.
- Auxiliar de ayuda a domicilio
- Auxiliar de clínica.
- Auxiliar de enfermería.
- Celador.
- Conductor.
- Cuidador.
- Director/gerente.
- Educador especial.
- Educador social.
- Equipo investigador.
- Fisioterapeuta.
- Geriatra.
- Logopeda.
- Médico general.
- Médico rehabilitador.
- Monitor de taller.
- Monitor ocio y tiempo libre.
- Neurocirujano.
- Neurólogo.
- Neuropsicólogo.
- Neuropediatra.
- Neuropsiquiatra.
- Nutricionista.
- Pedagogo.
- Personal ATS/DUE.
- Personal religioso y pastoral.
- Personal voluntario.
- Profesor o educador.
- Psicólogo.
- Psicopedagogo.
- Psicoterapeuta.
- Psiquiatra.
- Terapeuta de familia.
- Terapeuta ocupacional.
- Trabajador social.

4.4.1.3 Análisis del SGBD

La elección del SGBD más adecuado para la base de datos ocupa gran parte de los esfuerzos invertidos en la fase de análisis. Durante varios días se estuvo recopilando toda clase de información y referencias acerca de las alternativas existentes en el mercado en este sentido¹.

Por criterios económicos, se descartaron aquellos SGBD que exigían la adquisición de algún tipo de licencia para su uso. El estudio, por lo tanto, se centró únicamente en software gratuito, concretamente los SGBD investigados fueron Firebird, MySQL, PostgreSQL y SQLite.

MySQL se perfiló como la opción más acertada debido a su extensa comunidad de usuarios, la existencia de numerosa documentación que ayuda a su instalación y administración, y la potencia conseguida en la última versión lanzada (versión 5.0, que incluye disparadores, procedimientos almacenados e integridad referencial, entre otros).

4.4.2 Análisis de la aplicación Web de gestión de la base de datos

El análisis de los requisitos de usuarios referentes a la aplicación Web de gestión de la base de datos de la guía de recursos sobre daño cerebral adquirido, abarca temas como el tipo de usuarios de la misma, sus funcionalidades y forma de publicación. Antes de ello, se muestran los requisitos de usuario obtenidos tras la reunión con FEDACE.

4.4.2.1 Requisitos de usuario sobre la aplicación Web

Los objetivos a cumplir en el desarrollo de la aplicación Web de gestión a la base de datos son los siguientes:

- La aplicación Web debe estar disponible a través de Internet.
- La aplicación Web debe prestar especial atención a la accesibilidad.
- La administración de la aplicación Web será llevada desde FEDACE.

¹ Para más información consultar el capítulo 3, *Estado del Arte*.

- La aplicación Web de gestión de la base de datos no formará parte íntegra de la actual Web de FEDACE sino que será un sitio a parte. La estética e imagen corporativa de FEDACE debe ser igual o similar.
- Su acceso debe estar restringido a aquellos usuarios autorizados por FEDACE.
- El usuario administrador (FEDACE) podrá realizar todo tipo de acciones sobre la base de datos a través de la aplicación Web.
- El resto de usuarios sólo podrá consultar recursos y dejar comentarios acerca de los mismos visibles por otros usuarios.
- Debe permitir la posibilidad de cambiar el tipo de acceso para que cualquier usuario, autorizado o no, pueda únicamente consultar los recursos.
- El desarrollo y publicación de la aplicación Web tendrá el mínimo coste asociado.

4.4.2.2 Análisis de los tipos de usuario

Entre los requisitos de usuario se encuentra el que los usuarios autorizados por el administrador, FEDACE, tengan acceso a la consulta de los recursos disponibles además de la posibilidad de poder dejar escritos comentarios o evaluaciones personales acerca de estos recursos. Además, FEDACE desea que este acceso a las consultas pueda ser cambiado a un acceso libre en cualquier momento, manteniendo, eso sí, los privilegios existentes para los usuarios autorizados.

Por lo tanto, se identifican al menos tres tipos de usuario:

- El usuario administrador, con posibilidad de realizar todo tipo de operaciones (consultas y actualizaciones).
- El usuario autorizado, con posibilidad de consultar recursos y dejar comentarios sobre ellos.
- El usuario no autorizado, con permiso solamente para consultar recursos siempre y cuando el modo de acceso sea libre.

4.4.2.3 Análisis de las funcionalidades

La aplicación Web debe permitir una completa gestión de la base de datos, realizar consultas sobre diferentes criterios y permitir la existencia de diferentes usuarios con acceso a ella.

Por ello, las principales funcionalidades que debería contemplar son:

- Inserción de nuevos recursos.
- Borrado de los recursos ya existentes.
- Modificar cualquier aspecto de los recursos insertados.
- Consultar, bajo los criterios que se consideren más útiles por el desarrollador, los distintos recursos de la base de datos.
- Dar de alta o de baja a otros usuarios.
- Modificar los datos de acceso de los usuarios.
- Cambiar el modo de acceso a la aplicación Web.
- Restringir el acceso a ciertas funcionalidades dependiendo del perfil que se ostente.

4.4.2.4 Análisis de la accesibilidad Web

Uno de los requisitos más importantes sobre la aplicación Web de gestión de la base de datos es el que llegue a ser una Web plenamente accesible para aquellas personas que presenten alguna discapacidad. En este caso es especialmente importante y se comprende el énfasis de FEDACE en este aspecto puesto que una parte considerable de los futuros usuarios puede tener algún grado de discapacidad.

Para llegar al cumplimiento de este requisito, la herramienta más útil y que se debe seguir, al menos si se quiere cumplir con los estándares, es la recomendación dictada desde el W3C titulada “Pautas de Accesibilidad al Contenido en la Web” [27] y todos sus documentos asociados. En ellos se fijan las pautas que se deben seguir, casos específicos, soluciones posibles y pasos para llegar a una Web verdaderamente accesible.

En la verificación de dicho trabajo se pueden emplear algunas herramientas de validación automática existentes en Internet¹, que día a día mejoran en su funcionamiento y llegan a hacer, en algunos casos, análisis muy precisos sobre el nivel de accesibilidad de la Web procesada. Además, se deben realizar comprobaciones manuales sobre ciertos puntos para asegurar un análisis de la accesibilidad más fiable.

4.4.2.5 Publicación en Internet

El deseo de FEDACE es el de administrar y consultar la base de datos mediante una aplicación Web. El hecho de tener que conectarse a una base de datos, acceder a su información e integrarla en el código HTML de la página de forma dinámica, descarta la posibilidad de alojar la aplicación Web en alguno de los servidores de pago como, por ejemplo, FEDACE ha hecho hasta ahora con su página en Internet. Esto sería posible únicamente si la base de datos estuviera igualmente alojada en el servidor externo, pero para ello habría que dejar en manos ajenas el desarrollo de la base de datos y su puesta en funcionamiento, con el alto coste económico asociado. Otra posibilidad sería la conexión de una base de datos local con el servidor externo mediante un estándar ODBC, pero el proceso se complicaría bastante.

Es por ello que la mejor opción y más sencilla que se perfila es la de instalar un servidor propio en las oficinas de FEDACE. De esta forma, tanto la base de datos como la aplicación Web están administradas a nivel local, se tiene un mayor control sobre su funcionamiento y además se asegura una reducción de costes importante.

La puesta en marcha del servidor puede realizarse simplemente instalando tanto la base de datos como un software que haga de servidor Web en cualquiera de los equipos informáticos disponibles ya en FEDACE, pudiendo éste seguir utilizándose en las labores diarias de administración de la federación, al menos mientras el sistema completo siga siendo lo suficientemente ligero.

¹ Para más información consultar el capítulo 3, *Estado del Arte*

El servidor Web que se utilizará será el servidor Apache 2.0, debido a su amplia implantación en todo el mundo y la gran fiabilidad en cuanto a funcionamiento y seguridad que hasta ahora ha venido proporcionando¹.

En cuanto a la publicación en Internet, existen diferentes ofertas a la hora de contratar un dominio en la red, siendo muchas de ellas gratuitas a cambio de publicidad o incluso, en algunos casos, sin ella [37].

¹ Para más información consultar el capítulo 3, *Estado del Arte*.

5 DISEÑO DE LA BASE DE DATOS

Después del análisis efectuado sobre los requisitos de usuario expresados en la reunión mantenida con el cliente, llega la etapa de diseño. Ésta es sumamente importante puesto que define la estructura, comportamiento y futuro funcionamiento del sistema y proporciona los detalles para su correcta implementación. Cuanto mejor sea el diseño realizado, menos errores se cometerán en la parte de implementación y mayor nivel de satisfacción se alcanzará por parte del cliente.

En este capítulo se trata únicamente el diseño correspondiente a la base de datos, que junto con la aplicación Web, conforman el sistema completo sobre el que versa el presente proyecto.

5.1 Modelo conceptual: obtención del esquema E/R

El modelo conceptual constituye el primer paso en el diseño de una base de datos, y representa una modelización lógica del universo de discurso de forma que queden representados los conceptos fundamentales involucrados en el problema y las relaciones entre ellos, pero no ningún aspecto relacionado con la futura implementación física.

El modelo conceptual debe cumplir las siguientes características [Diseño de bases de datos relacionales, 1999]:

- Completo.
- Mínimo.
- Fácil de comprender.
- Fácil de formular.
- Formal, es decir, no ambiguo.
- Explícito en todas sus restricciones.

5.1.1 Construcción del modelo conceptual

En el modelo conceptual quedan reflejados elementos como las entidades, los atributos y las relaciones que pueden existir entre ellas, así como las características de estas relaciones.

Las entidades representan una unidad de significado dentro del universo de discurso, tienen existencia propia y diferenciada, y constituyen un elemento de interés específico para la resolución del problema. Según la definición del ANSI/SPAC (siglas en inglés de *American National Standard Institute, Standards Planning and Requirements Committee*) una entidad es “una persona, lugar, cosa, concepto o suceso, real o abstracto, de interés para la empresa” [Diseño de bases de datos relacionales, 1999].

Por otro lado, un atributo representa una propiedad o característica específica de una entidad concreta, o bien de una relación entre entidades. Un atributo se define sobre un dominio, que representa el conjunto de valores posibles para ese atributo.

Tras el análisis efectuado a los requisitos de usuario y, sobre todo, a los datos proporcionados por FEDACE sobre los recursos existentes, se han identificado para este universo de discurso las siguientes entidades, atributos y dominios.

➤ *RECURSO*

La entidad *RECURSO* representa a cada uno de los recursos existentes independientemente de su naturaleza, sean asociaciones o sean centros de atención de cualquier tipo. Esta entidad contempla una serie de atributos comunes a todos los recursos y son los siguientes:

- Nombre del recurso (identificador principal): nombre identificativo del recurso en cuestión.
- CIF (opcional): Código de Identificación Fiscal del recurso.
- Teléfono (opcional): atributo multivaluado para los teléfonos principales de contacto con el recurso.
- Fax (opcional): fax de contacto del recurso.
- E-mail (opcional): correo electrónico del recurso.
- Web (opcional): dirección de la página Web del recurso.

- Descripción (opcional): descripción general del recurso acerca de sus actividades, servicios ofrecidos y otra información adicional que se considere.

El dominio definido para cada uno de estos atributos queda reflejado en la siguiente tabla (Figura 5-1).

<i>Atributo</i>	<i>Dominio definido</i>
Nombre del recurso	cadena de entre 1 y 100 caracteres
CIF	cadena de 9 caracteres
Teléfono	entero de 9 dígitos
Fax	entero de 9 dígitos
E-mail	cadena de entre 1 y 50 caracteres
Web	cadena de entre 1 y 60 caracteres
Descripción	cadena de entre 1 y 65535 caracteres

Figura 5-1 Dominio de los atributos de la entidad *RECURSO*

➤ *ASOCIACIÓN*

Mediante la entidad *ASOCIACIÓN* se formaliza una particularización de la entidad *RECURSO*, y con ella se da representación a aquellos recursos que sean asociaciones de cualquiera de los dos tipos posibles, afectados o profesionales. Hereda todos los atributos de la entidad *RECURSO* y añade un único atributo:

- Presidencia (opcional): nombre del presidente, administrador principal o coordinador de la asociación de afectados o profesionales.

El dominio en el cual se define este atributo es el enseñado en la siguiente tabla (Figura 5-2):

<i>Atributo</i>	<i>Dominio definido</i>
Presidencia	cadena de entre 1 y 50 caracteres

Figura 5-2 Dominio de los atributos de la entidad *ASOCIACIÓN*

➤ *CENTRO DE ATENCIÓN*

La entidad *CENTRO DE ATENCIÓN* representa a aquellos recursos que son centros de atención de algún tipo. Es una particularización de la entidad *RECURSO* por lo que hereda todos sus atributos, además de incluir los siguientes:

- Dependencia (opcional): atributo multivaluado con el tipo de dependencia o titularidad del centro.
- Modelo de atención (opcional): atributo multivaluado que manifiesta los tipos de atención que ofrece el centro.
- Forma de acceso (opcional): los requisitos o criterios utilizados para la admisión de afectados en el centro.
- Perfil de usuario (opcional): el perfil o diagnóstico médico de los afectados que asisten al centro.

Los anteriores atributos están definidos sobre dominios indicados en la Figura 5-3.

<i>Atributo</i>	<i>Dominio definido</i>
Dependencia	enumerado con las posibles dependencias ¹
Modelo de atención	enumerado con los posibles modelos ¹
Forma de acceso	cadena de entre 1 y 65535 caracteres
Perfil de usuario	cadena de entre 1 y 65535 caracteres

Figura 5-3 Dominio de los atributos de la entidad *CENTRO*

La entidad *RECURSO* establece una relación de particularización con otras dos entidades, *ASOCIACIÓN* y *CENTRO*, a través del atributo vinculado a esa particularización “Tipo de recurso”. Este atributo contiene el tipo de recurso, ya sea “asociación de afectados”, “asociación de profesionales”, “centro de atención” o cualquier combinación de las tres. Esta relación entre las tres tablas se muestra gráficamente en la Figura 5-4.

¹ Consultar capítulo 4, apartado dedicado a *Análisis de los datos*

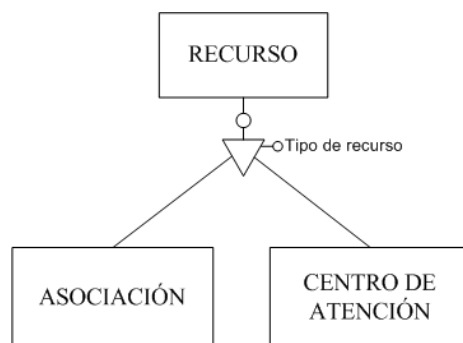


Figura 5-4 Particularización de la entidad *RECURSO*

La entidad *ASOCIACIÓN* representa a aquellos recursos que sean del tipo “asociación de profesionales” o “asociación de afectados”. Por otra parte, la entidad *CENTRO DE ATENCIÓN* recoge nuevos atributos para aquellos recursos que sean del tipo centro de atención. Tanto las asociaciones como los centros heredan, a través de la particularización, todos los atributos de la entidad *RECURSO*, pero además en cada caso añaden los atributos propios de su condición. Es decir, si por ejemplo se tiene una asociación de afectados o de profesionales, tendrá un nombre de recurso, un CIF, un teléfono, fax, correo electrónico, etc., y además de todos ellos tendrá los atributos propios de las asociaciones como es el nombre de la persona que ejerce la presidencia. Si fuera un centro, no sólo contendría los atributos del nombre del centro, el CIF, etc., sino que además sumaría otros como la dependencia del centro, el perfil del usuario, el modelo de atención o la forma de acceso.

Es importante resaltar que esta particularización es total y además no exclusiva, lo que quiere decir que cualquier recurso ha de ser obligatoriamente o bien asociación o bien centro de atención, y también se puede dar el caso de un recurso que sea a la vez asociación de cualquier tipo y centro de atención, teniendo en tal caso todos los atributos de las tres entidades *RECURSO*, *ASOCIACION* y *CENTRO DE ATENCIÓN*.

➤ *SEDE*

Se considera la posibilidad de que un mismo recurso, ya sea centro o asociación, pueda tener una o varias sedes teniendo cada una diferentes direcciones, datos de contacto, equipo médico, etc. Las sedes de un recurso se reflejan en esta entidad *SEDE* que contempla los siguientes atributos:

- Nombre de la sede (identificador principal): nombre que identifica de manera única a una sede dentro de un recurso determinado.
- Dirección completa: dónde se encuentra ubicada dicha sede. Es un atributo compuesto por los siguientes valores:
 - Provincia: provincia de la sede.
 - Localidad: localidad donde se ubica la sede.
 - Domicilio (opcional): tipo de vía (calle, plaza, avenida, etc.) y el número de ésta.
 - Código Postal (opcional): el código postal de la sede.
- Teléfono (opcional): atributo multivaluado para los teléfonos principales de contacto de la sede.
- Fax (opcional): fax de contacto de la sede.
- E-mail (opcional): correo electrónico de la sede.
- Web (opcional): página en Internet de la sede.
- Equipo: el equipo profesional del que dispone la sede. Es un atributo multivaluado y compuesto:
 - Cargo: cargo o puesto.
 - Número (opcional): número de profesionales que desempeñan ese cargo o puesto.
- Servicios (opcional): atributo multivaluado que expresa los servicios ofrecidos en la sede.
- N° de plazas o camas (opcional): el número de plazas o número de camas disponibles en la sede.
- Descripción (opcional): descripción detallada de cuál es el cometido de la sede, los servicios ofrecidos o el personal médico empleado en ella.

Los atributos de la entidad *SEDE* se definen sobre una serie de dominios descritos a través de la siguiente tabla (Figura 5-5)

<i>Atributo</i>	<i>Dominio definido</i>
Nombre de la sede	cadena de entre 1 y 80 caracteres
Provincia	cadena de entre 1 y 30 caracteres
Localidad	cadena de entre 1 y 50 caracteres
Domicilio	cadena de entre 1 y 50 caracteres
Código Postal	entero de 5 dígitos
Teléfono	entero de 9 dígitos
Fax	entero de 9 dígitos
E-mail	cadena de entre 1 y 50 caracteres
Web	cadena de entre 1 y 80 caracteres
Cargo	enumerado con los posibles cargos ¹
Número	entero de 5 dígitos
Servicios	enumerado con los posibles servicios ¹
Nº de plazas o camas	entero de 5 dígitos
Descripción	cadena de entre 1 y 65535 caracteres

Figura 5-5 Dominio de los atributos de la entidad *SEDE*.

Por lo tanto, una sede representa el lugar físico en el que se encuentra el recurso al que pertenece, donde está el equipo médico y donde se llevan a cabo los servicios ofrecidos. La entidad *SEDE* se relaciona con la entidad *RECURSO* mediante la relación *consta_de* (Figura 5-6). Esta relación constituye una dependencia en identificación y supone que la entidad *SEDE* es una entidad débil, puesto que necesita de la clave principal de la entidad *RECURSO* para estar plenamente identificada.

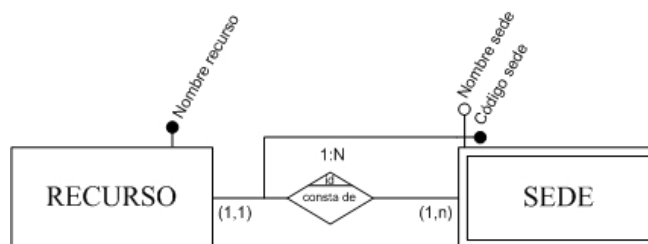


Figura 5-6 Relación *consta_de* entre *RECURSO* y *SEDE*

¹ Consultar capítulo 4, apartado dedicado a *Análisis de los datos*

Cada sede pertenecerá solamente a un único recurso y mismo recurso podrá tener N sedes diferentes. Todos los recursos han de tener al menos una sede. El motivo de que se haya separado un recurso de sus sedes es porque se necesitaba poder reflejar los casos en los que un mismo recurso estaba ubicado en diferentes lugares, teniendo cada uno una dirección propia, datos y personas de contacto propias, un equipo médico particular y una serie de servicios que puede que no se ofrezcan en otras sedes de ese mismo recurso. En caso de que un recurso tenga únicamente una sede no significa que en algún otro momento, siendo el recurso el mismo, pueda abrir otro local con servicios distintos a los que ofrecía en su primera localización. Es por ello que el equipo médico o los servicios no se han considera atributos o campos de un recurso en general, desde la entidad *RECURSO*, sino de una sede concreta de ese recurso dado. De esta manera, se puede obtener mejor la información sobre dónde, cómo y con qué equipo médico un recurso ofrece según qué clases de servicios.

➤ *CONTACTO*

La entidad *CONTACTO* refleja los datos de aquellas personas con las que se puede contactar dentro de un recurso dado, ya sean responsables médicos de algún área o personal de administración o de atención al público. Sus atributos son los siguientes:

- Nombre del contacto (identificador principal): nombre de la persona de contacto.
- Área (opcional): el área o responsabilidad que ocupa la persona de contacto dentro del recurso.
- Teléfono (opcional): teléfono del contacto.
- E-mail (opcional): correo electrónico del contacto

Los dominios siguientes (Figura 5-7) corresponden a la definición de estos atributos:

<i>Atributo</i>	<i>Dominio definido</i>
Nombre del contacto	cadena de entre 1 y 50 caracteres
Área	cadena de 1 y 30 caracteres
Teléfono	entero de 9 dígitos
E-mail	cadena de entre 1 y 50 caracteres

Figura 5-7 Dominio de los atributos de la entidad *CONTACTO*

La entidad *CONTACTO* se relaciona con la entidad *SEDE* mediante la relación *atendida_por*, mostrada en la Figura 5-8. Cada sede, por lo tanto, puede contar con una serie de personas de contacto propias de ella y no de otras sedes de ese mismo recurso. También es una entidad débil puesto que depende en identificación de la entidad *SEDE*.



Figura 5-8 Relación *atendida_por* entre *SEDE* y *CONTACTO*

Así mismo, un contacto pertenece a una única sede de un recurso dado, mientras que las sedes tienen N personas de contacto.

➤ *COMENTARIO*

Por último, los recursos tienen asociados una serie de comentarios sobre cada una de sus sedes, recogidos a través de la entidad *COMENTARIO*. Los atributos de esta entidad son los siguientes.

- Número de comentario (identificador principal): el número de comentario que representa entre todos los comentarios del recurso.
- Autor: el autor del comentario.
- Fecha: la fecha en la que se insertó el comentario sobre el recurso.
- Comentario: el comentario hecho sobre el recurso.

Los dominios de estos atributos se definen de la siguiente forma (Figura 5-9):

<i>Atributo</i>	<i>Dominio definido</i>
Número de comentario	entero de 4 dígitos
Autor	cadena de entre 1 y 20 caracteres
Fecha	fecha con formato aaaa/mm/dd
Comentario	cadena de entre 1 y 65535 caracteres

Figura 5-9 Dominio de los atributos de la entidad *COMENTARIO*

La entidad *COMENTARIO* se relaciona con la entidad *SEDE* a través de una relación llamada *tiene*, tal y como se indica en la Figura 5-10. Esta entidad representa una entidad débil con dependencia en identificación con la entidad *SEDE*. Un comentario pertenecerá únicamente a una sede de un recurso a la vez que las sedes pueden tener N comentarios.

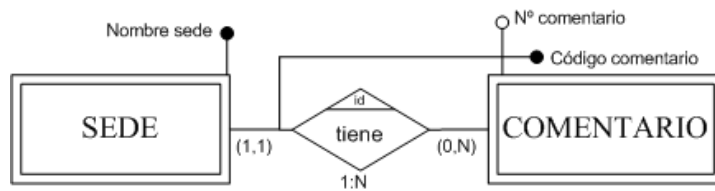


Figura 5-10 Relación *tiene* entre *SEDE* y *COMENTARIO*

5.1.2 Supuestos semánticos específicos

En este caso, no existe un enunciado del problema ni una explicación concreta de cómo se quiere que sea la base de datos, sino que se ha proporcionado una serie de datos que han de ser recogidos de la forma más eficiente y completa posible en una base de datos. El estudio y análisis de los propios datos, así como las reuniones con el cliente, ha sido lo que ha definido cuáles son los supuestos semánticos del problema:

- Un recurso puede ser una asociación de afectados, una asociación de profesionales o un centro de atención.
- En ocasiones los recursos son de más de un tipo al mismo tiempo.
- Dependiendo de su tipo, un recurso tiene una serie de datos asociados u otros.
- Un recurso puede tener diferentes direcciones o lugares donde realiza su actividad.
- En cada lugar puede contar con un equipo médico o servicios diferentes.
- Los recursos cuentan con un personal médico muy específico, teniendo profesionales de diferentes ramas médicas y pudiendo especificar un determinado número de ellos.
- Además de personal médicos, los recursos cuentan también con personal de otras áreas.
- Los recursos tienen personal encargado de ciertas áreas o de labores administrativas con el que se puede contactar.

- Los recursos ofrecen servicios de todo tipo a los afectados por daño cerebral y familiares de éstos.
- Los recursos pueden tener comentarios asociados que evalúen los servicios ofrecidos o la experiencia personal de los usuarios.

Además, los supuestos semánticos complementarios al esquema E/R son los siguientes:

- Se ha supuesto que todo recurso tiene al menos una sede.
- Se ha supuesto que las personas de contacto lo son únicamente de una sede.
- Se ha supuesto que las sedes pueden contar con sus propios teléfonos de contacto y fax.
- Se ha supuesto que las sedes pueden contar también con su propia dirección Web y un correo electrónico de contacto.
- Se ha supuesto que de la dirección física de una sede se conoce, al menos, la provincia y la localidad.
- Se ha optado por introducir un número identificativo y secuencial para los comentarios realizados sobre las sedes.

5.1.3 Diagrama E/R

Una vez explicadas las entidades, sus atributos, dominios e interrelaciones, se muestra en la siguiente figura (Figura 5-11) el diagrama E/R completo, fruto del modelado conceptual del universo de discurso del problema.

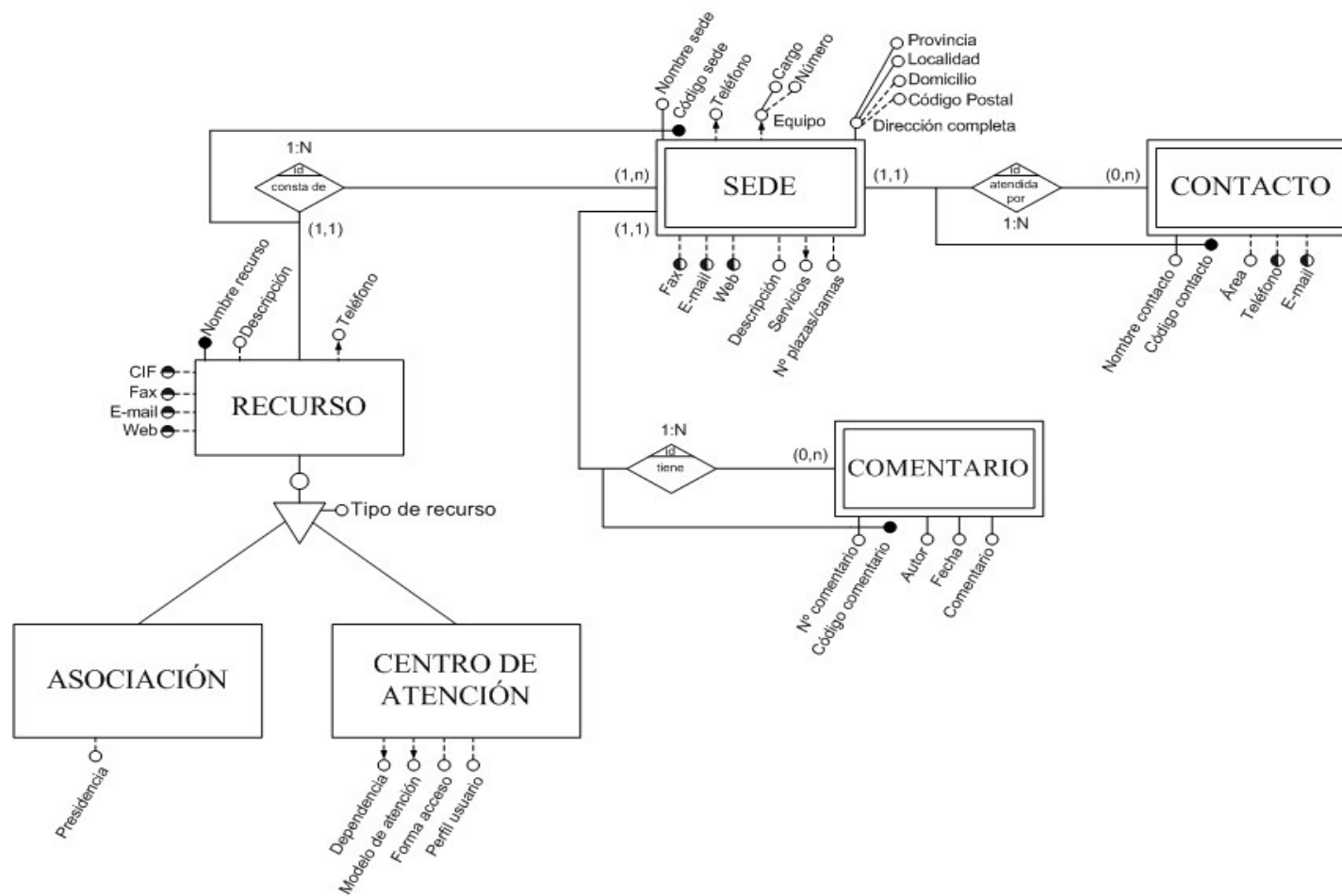


Figura 5-11 Diagrama E/R

5.1.4 Semántica no reflejada en el diagrama E/R.

No se ha podido reflejar:

- El tipo de un recurso puede ser “Profesionales”, “Afectados”, “Centro”, “Afectados y Profesionales”, “Profesionales y Centro”, “Afectados y Centro” o “Afectados, Profesionales y Centro”.
- El dominio definido para el campo “Servicio”¹.
- El dominio definido para el campo “Cargo”¹.
- El dominio definido para el campo “Modelo de atención”¹.
- El dominio definido para el campo “Dependencia”¹.
- Los campos “Teléfono” y “Fax” deben constar de 9 dígitos.
- El campo “Código postal” de estar formado por 5 dígitos.
- El campo “E-mail” de contener un símbolo arroba “@”.
- El campo “Provincia” debe contener el nombre de una provincia española.

5.2 *Diseño lógico estándar: obtención del esquema relacional*

El diseño lógico estándar de una base de datos proporciona una representación más cercana a la futura implementación física de diseño hecho sobre la base de datos. El esquema lógico estándar, correspondiente a esta base de datos, se puede derivar directamente del esquema E/R previamente analizado.

5.2.1 Transformación del esquema conceptual al esquema relacional

El esquema relacional se compone de dominios, relaciones, atributos e interrelaciones. Un dominio es un conjunto de valores de la misma naturaleza dentro del universo de discurso que afecta al problema.

Los atributos o columnas son propiedades que están definidas sobre un dominio y dan significado a una relación. Ésta pasa por ser una agrupación de atributos que constituyen juntos una unidad de significado. Las relaciones, también llamadas tablas, se

¹ Consultar capítulo 4, apartado sobre *Análisis de los datos*

materializan en tuplas o filas cuando los atributos que conforman la relación reciben un valor concreto.

Por otro lado, las relaciones se asocian entre sí mediante las interrelaciones, que llevan siempre asociadas una cardinalidad.

A continuación, se explica el proceso completo de transformación del esquema conceptual o esquema E/R anteriormente construido en el esquema relacional.

5.2.1.1 Transformación de entidades

Este paso se basa en una sencilla regla, y es que cada tipo de entidad del esquema E/R se transforma en una nueva relación en el esquema relacional. De esta forma, cada una de las entidades analizadas en el esquema E/R (*RECURSO*, *ASOCIACIÓN*, *CENTRO DE ATENCION*, *SEDE*, *CONTACTO* y *COMENTARIO*) derivan en relaciones con el mismo nombre en el actual esquema relacional.

5.2.1.2 Transformación de los atributos

La transformación de los atributos de las entidades del esquema E/R en nuevos atributos de las relaciones en esquema relacional se realiza cumpliendo las siguientes reglas:

1. El identificador principal de la entidad se transforma en la clave principal en la relación derivada. Así, las nuevas relaciones quedan como indica la Figura 5-12

```
RECURSO ( Nombre_recurso )  
ASOCIACION ( Nombre_recurso )  
CENTRO ( Nombre_recurso )  
SEDE ( Nombre_sede )  
CONTACTO ( Nombre_contacto )  
COMENTARIO ( Num_comentario )
```

Figura 5-12 Claves principales

2. Los identificadores alternativos de la entidad se transforman en claves alternativas en la relación derivada. Las relaciones que cuentan con atributos que son claves alternativas se representan con un subrayado discontinuo y son las siguientes (Figura 5-13):

RECURSO (Nombre_recurso, CIF, Telefono1, Telefono2, Telefono3, Fax, E-mail,
Web)

SEDE (Nombre_sede, Telefono1, Telefono2, Telefono3, Fax, E-mail, Web)

CONTACTO (Nombre_contacto, Telefono, E-mail)

Figura 5-13 Claves alternativas

El motivo por el cual el atributo multivaluado “Teléfono” de las entidades *RECURSO* y *SEDE*, que aparecían en el esquema E/R, se haya transformado en tres atributos diferentes, “Teléfono1”, “Teléfono2” y “Teléfono3”, es porque para el cliente era muy importante poder realizar búsquedas sobre recursos o sus sedes mediante los teléfonos de información de los mismos. Así pues, previo pacto con el cliente, se hace necesario dar una mayor relevancia a estos campos convirtiéndolos en claves alternativas para facilitar su indexación y acceso a los datos.

3. Los atributos obligatorios y opcionales de la entidad se transforman en atributos obligatorios y opcionales en la relación derivada, identificando mediante un asterisco (“*”) a los últimos.
4. Los atributos multivaluados de la entidad se transforman en una nueva relación formada por el identificador principal de la entidad como clave principal y el atributo multivaluado, siendo ambos la clave primaria de la nueva relación. En este caso, los atributos multivaluados que existían en el esquema E/R eran “Dependencia” y “Modelo de atención” en *CENTRO DE ATENCIÓN* (mostrados en la Figura 5-14), y los atributos “Servicio” y “Equipo” en *SEDE* (mostrados en la Figura 5-15).

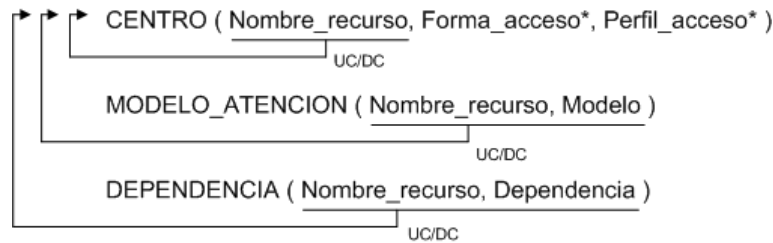


Figura 5-14 Transf. de los atributos multivaluados *CENTRO DE ATENCIÓN*

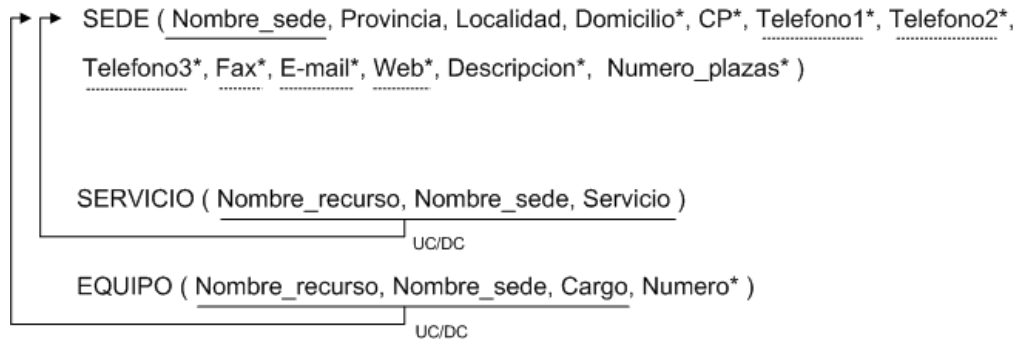


Figura 5-15 Transformación de los atributos multivaluados de la entidad *SEDE*

5.2.1.3 Transformación de dependencias en identificación

En el caso de existir dependencias en identificación, éstas se transforman en el esquema relacional mediante la propagación de clave desde la entidad fuerte hasta la entidad débil. En la relación que corresponde a la entidad débil dentro del esquema E/R, la clave principal se forma con la unión de la clave principal de la relación correspondiente a la entidad fuerte (llamada “clave ajena”) y la clave principal de la entidad débil.

El esquema E/R propuesto contiene 3 dependencias en identificación, que serían transformadas de la forma siguiente:

1. Dependencia en identificación entre la entidad *RECURSO* y la entidad *SEDE* (Figura 5-16).

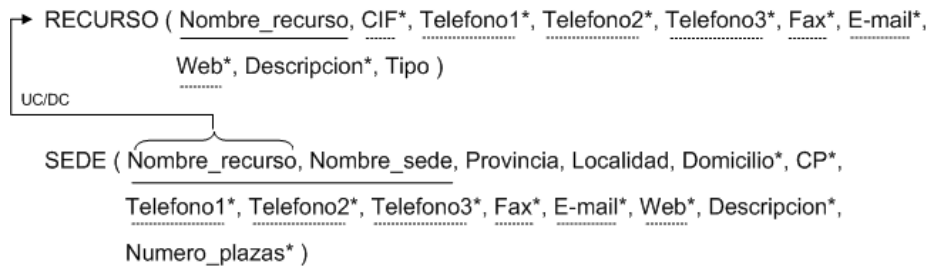


Figura 5-16 Transformación de la dependencia en identificación de *SEDE*

2. Dependencia en identificación entre la entidad *SEDE* y la entidad *CONTACTO* (Figura 5-17).

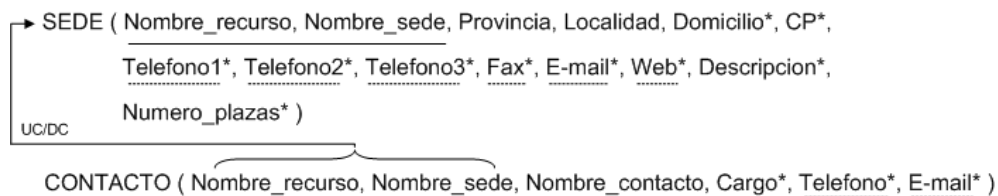


Figura 5-17 Transformación de la dependencia en identificación de *CONTACTO*

3. Dependencia en identificación entre la entidad *SEDE* y la entidad *COMENTARIO* (Figura 5-18).

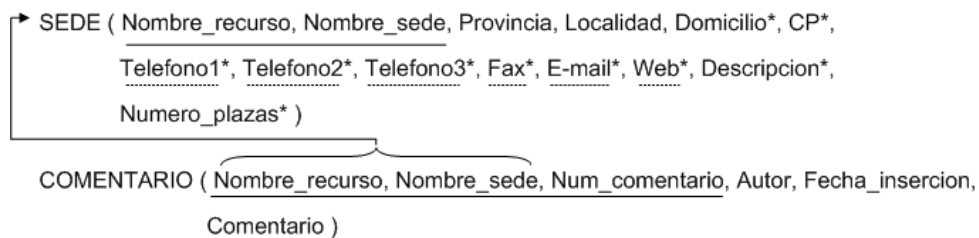


Figura 5-18 Transformación de la dependencia en identificación de *COMENTARIO*

5.2.1.4 Transformación de tipo y subtipos

Existen diferentes formas de transformar una particularización de una entidad en función del valor del atributo discriminante de dicha particularización. En este caso, se opta por convertir cada subtipo o particularización en una relación diferente mediante una clave ajena, tal y como lo indica la Figura 5-19, ya que cada uno de los subtipos, *CENTRO* y *ASOCIACIÓN*, cuenta con sus propios atributos.

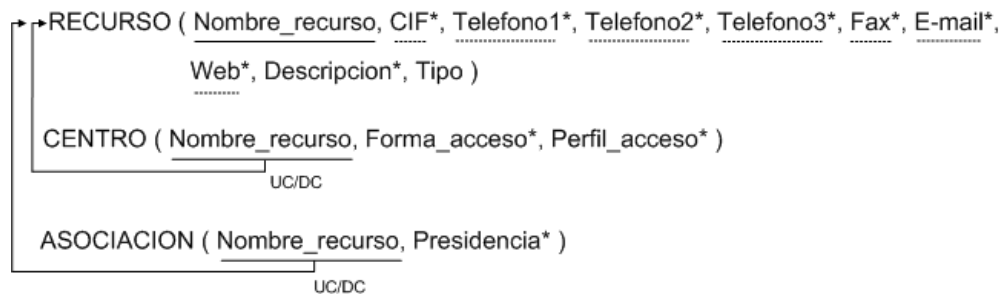


Figura 5-19 Transformación subtipos *ASOCIACIÓN* y *CENTRO DE ATENCIÓN*

5.2.2 Diagrama relacional

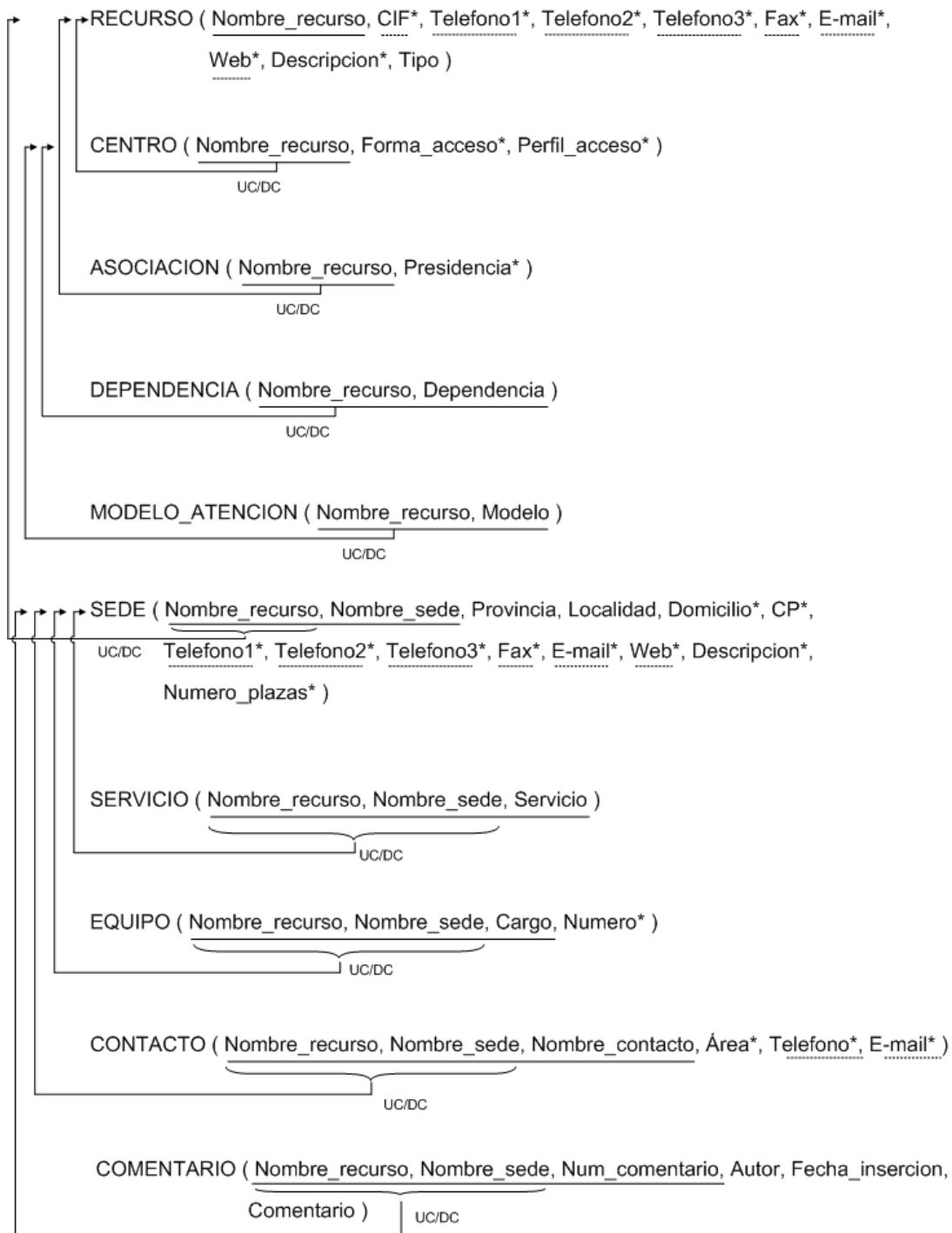


Figura 5-20 Diagrama relacional

5.3 Semántica no reflejada en el diagrama relacional

Para recoger toda la semántica requerida para una fiel representación del universo de discurso del problema y no contemplada a través del diagrama relacional anteriormente mostrado (Figura 5-20), se hace necesario introducir algunos elementos como las comprobaciones (*checks*), los disparadores (*triggers*) y aserciones (*assertions*).

5.3.1 Comprobaciones derivadas de los dominios de los atributos

A continuación se enumeran comprobaciones que deben realizarse sobre algunos atributos de ciertas tablas para respetar su dominio. El SGBD utilizado, MySQL, admite la implementación de *checks* pero actualmente los ignora por completo, siendo necesario realizarlos mediante disparadores o procedimientos almacenados. Deben realizarse las siguientes comprobaciones:

- Los campos “Telefono1”, “Telefono2”, “Telefono3” y “Fax” de las tablas *RECURSO* y *SEDE*, al igual que el campo “Telefono” de la tabla *CONTACTO*, deben tener 9 dígitos.
- El código postal de la tabla *SEDE* debe constar de 5 dígitos.
- El campo “Dependencia” de la tabla *DEPENDENCIA* sólo puede ser de alguno de los siguientes valores: “Público”, “Privado concertado”, “Privado lucrativo”, “Privado no lucrativo”.
- El campo “Modelo” de la tabla *MODELO_ATENCION* deberá reflejar sólo los valores permitidos¹.
- El campo “Servicio” de la tabla *SERVICIO* contendrá sólo un rango de valores determinado¹.
- El campo “Cargo” de la tabla *EQUIPO* contendrá solamente alguno de los valores permitidos¹.
- El campo “Provincia” de la tabla *SEDE* tomará el valor de alguna de las provincias españolas.
- El correo electrónico debe tener una arroba ‘@’.

¹ Consultar capítulo 4, apartado sobre *Análisis de los datos*.

5.3.2 Semántica perdida en la transformación del diagrama E/R al relacional

Existe una cierta pérdida de semántica derivada de la transformación del diagrama E/R al diagrama relacional. En el problema que ocupa este proyecto, esta pérdida de semántica se manifiesta en la transformación de la jerarquía de particularización y en la cardinalidad mínima que debe existir en las relaciones.

5.3.2.1 Transformación de jerarquías

La transformación de la particularización existente entre la entidad *RECURSO* y las entidades *ASOCIACIÓN* y *CENTRO* en tres relaciones de igual nombre en el esquema relacional implica una serie de comprobaciones para reflejar la completa semántica de esta jerarquía.

- Por un lado, comprobar mediante un *check* que el campo “Tipo” de la relación *RECURSO* contiene alguno de los siguientes posibles valores:
 - “Profesionales”
 - “Afectados”
 - “Centro”
 - “Afectados y Profesionales”
 - “Profesionales y Centro”
 - “Afectados y Centro”
 - “Afectados, Profesionales y Centro”
- Por otro lado, controlar que, efectivamente, cada recurso insertado en la relación *RECURSO* tiene su correspondiente tupla en las tablas *ASOCIACIÓN*, *CENTRO*, o en ambas, dependiendo siempre del valor existente en el campo “Tipo” de *RECURSO*. Esta comprobación debe realizarse a través de una aserción (*assertion*).

5.3.2.2 Cardinalidad mínima

La cardinalidad es otra comprobación que debe realizarse de forma expresa debido a su pérdida por la transformación del diagrama E/R al relacional. Esta cardinalidad mínima se comprueba con el uso de aserciones (*assertions*).

La cardinalidad mínima existente entre las entidades *RECURSO* y *SEDE* exige que cada recurso tenga al menos una sede. La forma de implementar esta condición mediante una aserción debería ser controlando que cada nombre de recurso presente en el campo “Nombre recurso” de la tabla *RECURSO* esté presente también y al menos una vez en el mismo campo, “Nombre recurso”, de la tabla *SEDE*.

6 DISEÑO DE LA APLICACIÓN WEB

Para diseñar correctamente una aplicación Web se deben tener en cuenta ciertos principios básicos de lo que ha pasado a llamarse usabilidad Web. Usabilidad es una medida de lo fácil, rápido y agradable que es utilizar una Web con un determinado fin, ya sea laboral, de ocio o meramente informativo. Para cubrir las partes del diseño Web basadas en la usabilidad, se hará uso de algunas recomendaciones recogidas en diferentes publicaciones sobre el tema [Usabilidad de sitios Web, 2002], [Diseño de páginas Web, 1997].

Pero antes de saber qué tipo de diseño se le quiere proporcionar a la aplicación, se deben plantear previamente una serie de cuestiones. Por ejemplo, cuál es el motivo que lleva a plantearse el desarrollo de la misma o, en otras palabras, qué objetivos se pretenden conseguir. Otro tema importante a abordar es a quién está destinada, si existe un grupo de usuarios muy concreto que son a los que se quiere atraer o si la aplicación Web estará dirigida a toda clase de público. Algunos autores hacen una gran diferenciación entre dos tipos de aplicaciones Web: aquellas que tienen un objetivo comercial y buscan un beneficio económico, y aquellas que tienen un objetivo puramente informativo sobre uno o varios temas [Diseño de páginas Web, 1997]. Si se aceptase esta clasificación, evidentemente se estaría ante una aplicación del segundo tipo.

En las siguientes secciones se desarrollan diferentes aspectos del diseño de la aplicación Web de gestión de la base de datos de FEDACE.

6.1 Funcionalidades de la aplicación Web

El objetivo principal de la aplicación Web es ofrecer un entorno gráfico y sencillo para realizar las labores básicas de gestión de la base de datos. Tareas como insertar, borrar o modificar datos deberían poder realizarse de forma rápida e intuitiva. Otras funciones, como la consulta de cualquier recurso bajo un abanico amplio de criterios de búsqueda también debería realizarse mediante el uso de la aplicación. Al existir una serie de usuarios a los que se les autoriza a acceder a la aplicación Web mediante un nombre de

usuario y una contraseña, requiere además añadir la posibilidad tanto de dar de alta a nuevos usuarios como dar de baja a usuarios ya existentes. En cualquier momento, según deseos de FEDACE, se debe permitir acceder a la aplicación Web a cualquier persona que lo desee, es decir, poder cambiar el modo de acceso de una forma restringida a otra sin ningún tipo de traba.

Analizando los requisitos del usuario, así como el diseño final de la base de datos (sus tablas, atributos, etc.) se han identificado las siguientes funcionalidades:

- **[Funcionalidad 1] Insertar un recurso:** rellenar en un formulario Web todos aquellos campos necesarios para insertar un nuevo recurso en la base de datos. No sólo entrarán en juego los atributos de la tabla *RECURSO* (unidos a los atributos de sus tablas asociadas *CENTRO*, *ASOCIACIÓN*, *MODELO DE ATENCIÓN* y *DEPENDENCIA*) si no también los de la tabla *SEDE* (junto a los atributos de las tablas *SERVICIO* y *EQUIPO*) puesto que existe la restricción de que todo recurso ha de tener al menos una sede donde está ubicado.
- **[Funcionalidad 2] Insertar una sede:** un recurso, aunque deba disponer de una única sede, puede constar de varias de ellas. De esta forma, una vez insertado un recurso, se debe brindar la posibilidad de añadir nuevas sedes vinculadas a este recurso. Las tablas *SEDE*, *SERVICIO* y *EQUIPO* estarán involucradas en todo el proceso de inserción de una nueva sede, que se llevará a cabo rellenando el correspondiente formulario Web.
- **[Funcionalidad 3] Insertar un contacto:** las sedes, como se ha explicado anteriormente, cuentan con una serie de personas con las que se puede contactar, ya sean personal médico responsable de un área concreta o meramente administrativo. Una vez insertada una sede de un recurso, a través de esta opción se podrá añadir un contacto vinculado a esa sede rellenando un formulario. La tabla *CONTACTO* con todos sus atributos forman parte de esta funcionalidad.

- **[Funcionalidad 4] Insertar un comentario:** los diferentes usuarios autorizados pueden hacer comentarios o dejar mensajes acerca de su experiencia personal en cada uno de los recursos presentes en la base de datos. Una vez insertado un recurso, a través de esta opción se podrán realizar, rellenando el correspondiente formulario Web, comentarios sobre cualquier de sus sedes a través de inserciones en la tabla *COMENTARIO*.

- **[Funcionalidad 5] Modificar un recurso:** cuando se ha insertado un recurso, puede surgir la necesidad de cambiar cualquiera de sus datos generales, bien porque éstos hayan cambiado o bien porque se ha cometido un error durante su inserción. Mediante esta opción se permitirá cambiar estos datos editándolos sobre un formulario Web que contendrán los datos originariamente introducidos. Únicamente se modificarán los que están contenidos en las tablas *RECURSO* y las relacionadas a ésta como *ASOCIACIÓN*, *CENTRO*, *MODELO DE ATENCIÓN* y *DEPENDENCIA*. Las sedes, contactos o comentarios serán editados desde otras opciones.

- **[Funcionalidad 6] Modificar una sede:** cualquiera de las sedes de un recurso podrá ser modificada a través de esta opción, donde se visualizará un formulario Web con los datos actuales de la sede. Los datos que aquí se podrán modificar serán aquellos relacionados únicamente con lo que son los datos generales de una sede, sin tener en cuenta los contactos ni los comentarios. Así, las tablas vinculadas serán *SEDE*, *SERVICIO* y *EQUIPO*.

- **[Funcionalidad 7] Modificar un contacto:** las personas de contacto de una sede determinada podrán ser modificadas mediante esta opción mediante un formulario Web con los datos insertados previamente. En él estarán presentes todos los atributos de la tabla *CONTACTO*.

- **[Funcionalidad 8] Modificar un comentario:** el usuario con el rol de administrador de la base de datos deberá tener la posibilidad de editar los comentarios dejados por otros usuarios. Mediante esta opción se visualizará en una caja de texto editable el comentario seleccionado y quedará modificado de forma permanente en la tabla *COMENTARIO*.

- **[Funcionalidad 9] Consultar un recurso:** el objetivo principal de los usuarios que se conecten a la aplicación Web será el de consultar los recursos presentes dentro de la base de datos. Es por ello que se debe prestar especial atención a los criterios de búsqueda para que estos usuarios encuentren de la forma más rápida y eficiente posible los recursos que están buscando. Los criterios de búsqueda establecido serán los siguientes:
 - Nombre del recurso
 - Tipo de recurso
 - Dependencia
 - Modelo de atención
 - Comunidad autónoma y/o provincia
 - Servicios
 - Teléfono

- **[Funcionalidad 10] Consultar comentarios:** el usuario que hace de administrador de la base de datos podrá consultar aquellos comentarios hechos por el resto de usuarios sobre un recurso dado. Esta consulta permitirá controlar de manera más eficiente aquellos comentarios que no se ajusten al criterio de FEDACE (comentarios ofensivos, poco claros, de dudosa utilidad, etc.)

- **[Funcionalidad 11] Consultar usuarios:** el usuario administrador podrá consultar mediante esta opción a los demás usuarios autorizados a acceder a la aplicación Web, además de conocer sus nombres de usuario y contraseñas, entidades a las que pertenecen (otras asociaciones, centros, etc.) y otra información introducida por el propio administrador.

- **[Funcionalidad 12] Dar de alta a un usuario:** el usuario que asuma el papel de administrador de la base de datos podrá dar de alta a otros usuarios para que puedan acceder a la aplicación Web mediante un nombre de usuario y una contraseña que éste les proporcionará. El usuario nuevo quedará reflejado en la tabla *USUARIO*.

- **[Funcionalidad 13] Dar de baja a un usuario:** los usuarios de la aplicación Web podrán ser dados de baja por el usuario administrador, de forma que ya no podrán realizar consultas ni dejar comentarios sobre los recursos. El usuario será borrado de la tabla *USUARIO*.

- **[Funcionalidad 14] Cambiar contraseña:** cualquier usuario, sea del tipo que sea, puede cambiar su contraseña proporcionando para ello la contraseña que actualmente tiene asignada.

- **[Funcionalidad 15] Cambiar el modo de acceso:** el usuario con el rol de administrador podrá cambiar el modo de acceder a la aplicación Web por parte del resto de usuarios. Los dos modos de acceso serán, por un lado, mediante la introducción de nombre de usuario y contraseña para los usuarios previamente registrados y, por otro lado, acceso completamente libre a la aplicación. Cuando se establezca un acceso libre a la aplicación Web, aparecerá en la página principal de identificación un enlace que llevará directamente a las consultas sobre los recursos de la base de datos, no haciendo falta proporcionar ni nombre de usuario ni contraseña de ningún tipo. La forma de conectar a la aplicación Web para los usuarios registrados se seguirá manteniendo bajo este acceso libre, al igual que todos sus permisos asociados.

6.2 Perfiles de usuario

Los usuarios que se conecten a la aplicación Web de gestión de la base de datos tendrán diferentes permisos y posibilidades de consulta sobre la misma en función del perfil que ostenten. Según los requisitos manifestados por FEDACE, se identifican los siguientes tipos de usuarios dentro de la aplicación Web:

- **Usuario Administrador**
Las funciones del usuario “Administrador” estarán desempeñadas por el personal de FEDACE y dispondrán de una única cuenta, con nombre y clave de acceso, con el que poder conectarse a la aplicación bajo este perfil. El usuario “Administrador” será el encargado de mantener actualizada la base de datos,

teniendo permiso para insertar, borrar o modificar cualquiera de los elementos pertenecientes a la misma (recursos, sedes, personas de contacto, etc.). Los comentarios escritos por los demás usuarios pueden ser editados por el usuario “Administrador” si así lo considera necesario, e incluso borrados si juzga su contenido como ofensivo, inapropiado o de escaso interés informativo. Además de ello, el usuario “Administrador” es el único que decide sobre la política de acceso a la aplicación Web. Parte de su responsabilidad consistirá en formalizar las altas y bajas de nuevos usuarios, proporcionándoles un nombre de usuario (*login*) y una contraseña de acceso inicial (*password*) con el que poder conectarse a la aplicación Web. Así mismo, parte de esta política de acceso consiste en establecer cuando así lo desee un acceso restringido a la aplicación Web (únicamente usuarios registrados) o un acceso libre (cualquier persona puede acceder, conservando los usuarios registrados todos sus privilegios)

- **Usuario Registrado**

Un usuario registrado es aquél que ha sido dado de alta previamente por el administrador de la aplicación Web. Este usuario recibirá, por la vía que considere el usuario “Administrador”, un nombre de usuario y una clave de acceso a la aplicación. El usuario “Registrado” dispondrá de permisos para consultar cualquier recurso de la base de datos mediante el formulario de consulta de la aplicación Web. También tendrá la posibilidad de dejar comentarios sobre cualquiera de los recursos, a modo de evaluación o exposición de su experiencia personal con ellos. Estos comentarios serán visibles por todos los demás usuarios, aunque será el usuario “Administrador” el que decida si ese comentario es apropiado o si debe ser modificado o borrado. Por último, el usuario “Registrado” podrá, en todo momento, cambiar su clave de acceso por una nueva.

- **Usuario No Registrado**

El usuario “No Registrado” es todo aquél usuario que no es del tipo “Registrado” o “Administrador” y que accederá a la aplicación Web únicamente cuando en ésta se haya establecido el acceso libre por el usuario “Administrador”. Cuando el acceso a la aplicación Web sea libre, aparecerá en

la pantalla inicial de identificación un enlace directo a través del cual cualquier usuario puede acceder de forma directa a las consultas. Consultar recursos, incluidos sus comentarios, es la única opción disponible para este tipo de usuarios. Los usuarios “Registrados” conservarán sus privilegios (consulta de recursos e inserción de comentarios) y podrán acceder a la aplicación Web identificándose previamente con su nombre de usuario y contraseña, es decir, de la misma forma que lo harían bajo el acceso restringido.

En la siguiente tabla, puede apreciarse un resumen de cuáles son los permisos que cada uno de los posibles usuarios de la aplicación Web lleva asociados (Figura 6-1).

	Administrador	Registrado	No Registrado
Func 1	Sí	No	No
Func 2	Sí	No	No
Func 3	Sí	No	No
Func 4	Sí	Sí	No
Func 5	Sí	No	No
Func 6	Sí	No	No
Func 7	Sí	No	No
Func 8	Sí	No	No
Func 9	Sí	Sí	Sí
Func 10	Sí	No	No
Func 11	Sí	No	No
Func 12	Si	No	No
Func 13	Sí	No	No
Func 14	Sí	Sí	No
Func 15	Sí	No	No

Figura 6-1 Usuarios y permisos asociados

6.3 Navegabilidad de la aplicación Web

La navegabilidad representa la relación entre las diferentes páginas de una aplicación Web a través de vínculos o enlaces de hipertexto que permiten al usuario moverse entre ellas y encontrar, de una forma más o menos lógica, aquello que está buscando.

Básicamente, se debe ayudar al usuario a contestar de forma rápida a tres cuestiones fundamentales: dónde se ha estado, dónde se está y a dónde se puede ir. Existen una serie de consejos básicos a la hora de diseñar la forma de navegar entre las páginas de una aplicación Web [Diseño de páginas Web, 1997]:

- Tener siempre visible un menú o barra de navegación con todas las páginas del sitio o al menos con aquellas que sean las principales.
- Se debe poder acceder a la página principal desde cualquiera de las otras páginas de la Web.
- Debe ser posible acceder a través de algún enlace a cualquiera de las páginas que forman parte de un sitio Web. Una página situada en “vía muerta”, es decir, a la que no se puede volver, es uno de los errores más molestos para el usuario.
- Diseñar la Web de forma que no sea necesario tener que utilizar los botones “Anterior” y “Siguiente” del navegador.
- Si una sección contiene otras páginas relativamente importantes, es conveniente hacer visible en todo momento otro menú con enlaces a cada una de ellas.
- Los enlaces deben ser visibles, claros y descriptivos.
- Tratar que lo primero que se cargue de cualquier página sean los enlaces o barras de navegación, de forma que el usuario pueda cambiar de página sin esperar a que se carguen por completo todos sus contenidos.
- Intentar seguir la regla de los tres “clicks”: una página debe ser accesible desde cualquier otra haciendo “click” en tres enlaces a lo sumo.

Siguiendo el criterio marcado por estos consejos, además de por la propia experiencia del autor del presente proyecto en la navegación Web, se ha considerado oportuno agrupar todas las funcionalidades de la aplicación Web de gestión de la base de datos entorno a estas opciones principales:

- **Inicio:** para volver a la página principal de la aplicación Web, donde aparece el mensaje de bienvenida y la guía de accesibilidad.
- **Consultar:** para realizar consultas sobre recursos (a las que tendrán acceso todos los usuarios), así como sobre los comentarios y los usuarios registrados de la aplicación Web (sólo el usuario “Administrador”).
- **Gestión de usuarios:** para dar de alta a un nuevo usuario o dar de baja a un usuario ya existente (usuario “Administrador”).
- **Actualizar:** para realizar todas las tareas de inserción, borrado y modificación sobre los elementos de la base de datos (usuario “Administrador” y usuario “Registrado” para la inserción de nuevos comentarios).
- **Control de acceso:** para establecer el modo de acceso a la aplicación Web, pudiendo ser un acceso restringido o un acceso libre (sólo usuario “Administrador”).
- **Cambio de contraseña:** para cambiar la clave de acceso a la aplicación Web (tanto usuario “Administrador” como usuario “Registrado”).
- **Salir:** para finalizar la sesión actual en la aplicación Web.

A continuación se muestra un diagrama de navegación (Figura 6-2) donde se puede comprobar el acceso a cada página de la aplicación Web según el perfil de usuario.

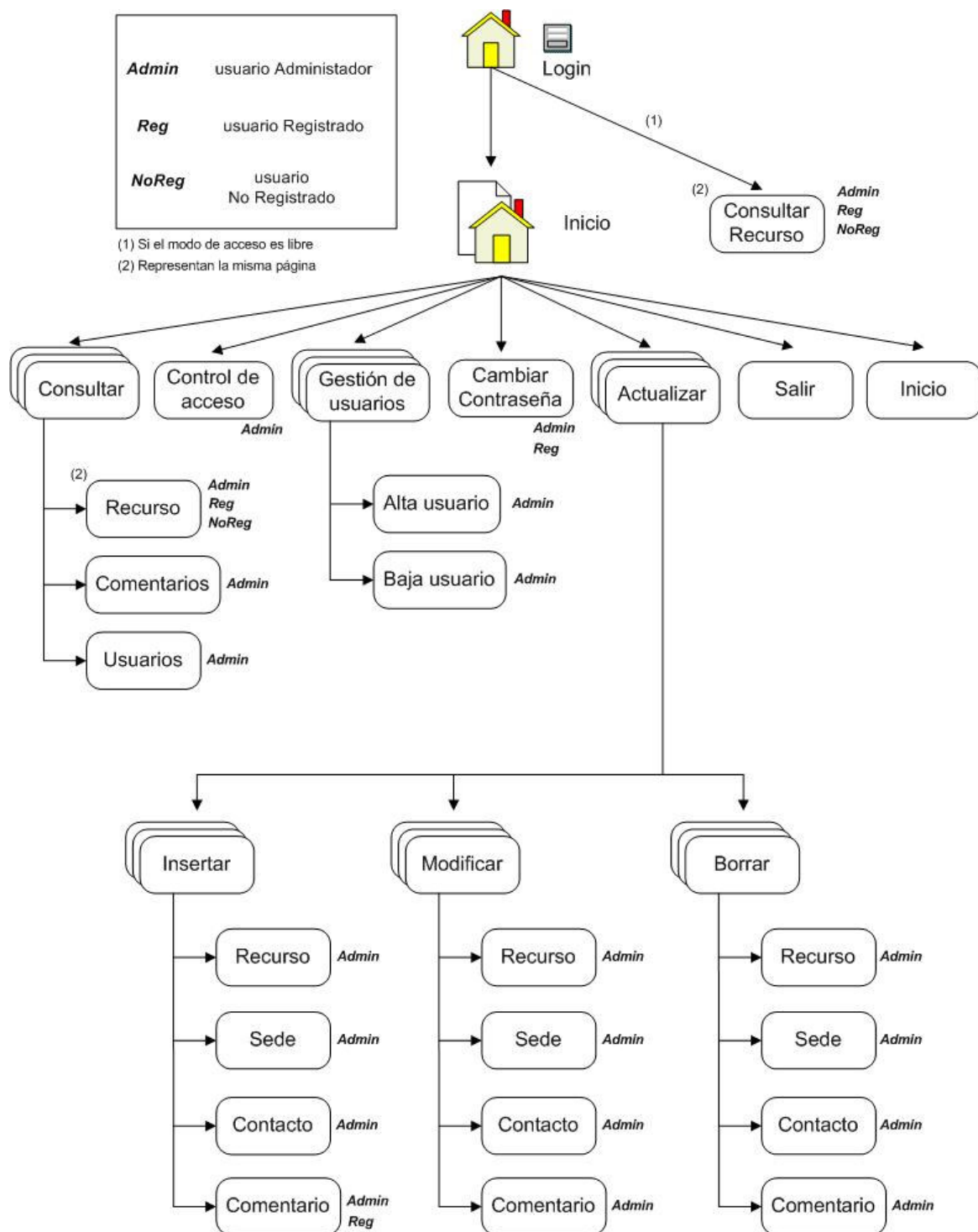


Figura 6-2 Diagrama de navegación de la aplicación Web

6.4 Interfaz de la aplicación Web

La interfaz representa aquel componente del software que facilita la comunicación con el usuario y permite el flujo de información en ambos sentidos, por lo que su correcto diseño es un punto muy sensible al que se debe prestar la debida atención. Una interfaz Web debe ser sencilla pero eficaz al mismo tiempo.

Además, la interfaz constituye parte del formato o el aspecto del contenido, y ha de estar siempre separado de la estructura de éste. Las peculiaridades de la interfaz de una aplicación Web generalmente se proporcionan a través del uso de hojas de estilo o CSS.

6.4.1 Distribución de la pantalla

La distribución por la pantalla de los elementos que se desea que aparezcan en la aplicación Web debe hacerse de forma que no molesten ni distraigan la atención del usuario y le sirvan de guía para encontrar fácil y rápidamente aquella información que se busca. Los elementos de navegación o de índice de contenidos deben orientarse ante todo a la simplicidad, con las mínimas distracciones posibles y con una arquitectura de la información muy clara.

Se hace mucho hincapié por parte de los desarrolladores de aplicaciones Web de la importancia de disponer de una barra de navegación o menú con las principales secciones o páginas de la aplicación Web con el fin de permitir al usuario acceder en cualquier momento y de forma directa a aquellas partes que más le interesen. La aplicación Web de gestión de la base de datos, tal y como se ha comentado en la sección anterior, se divide en varias opciones o funcionalidades principales que, en algunos casos, pueden contar con otras funcionalidades que dependen de la principal. Por ello, se decidió hacer visible en todo momento dos barras de navegación distintas. Por un lado, la barra de navegación principal, situada de forma horizontal en la parte superior de la página y que contiene las principales secciones de la aplicación Web:

- Inicio
- Consultar
- Actualizar

- Gestión de usuarios
- Control de acceso
- Cambio de contraseña
- Salir

Por el otro lado, disponer igualmente de una barra de navegación secundaria que, en función de la opción principal en la que el usuario se encuentre, muestra las diferentes opciones dentro de la principal. Esta barra de navegación secundaria se sitúa en la parte izquierda de la pantalla de forma horizontal. En la Figura 6-3 puede apreciarse la colocación de estos elementos.



Figura 6-3 Barras de navegación principal y secundaria

Las barras de navegación constituyen quizás el componente más importante de los presentes en la interfaz, pero no son el único. Además, en la parte superior derecha, se sitúa el logotipo de FEDACE que sirve de enlace con la Web de la que dispone dicha federación y está siempre presente en la interfaz de la aplicación Web para la gestión de la base de datos.

El título de la aplicación Web, “Guía de Recursos sobre Daño Cerebral Adquirido” es el primer elemento que el usuario visualiza una vez cargada la página y pretende ser lo suficientemente explicativo para que el usuario sepa a simple vista, en caso de ser la

primera vez que se conecta, cuál es el cometido de esta Web. Ese mismo encabezado constituye el título de la aplicación Web según el etiquetado HTML (mediante el elemento TITLE) y es de suma importancia puesto que define a toda la aplicación en su conjunto. Muchos diseñadores de páginas Web eligen títulos equivocados o aprovechan ese espacio (situado en la barra superior de los navegadores) para incluir mensajes de todo tipo. Esto es un error y desaprovecha el potencial del título de una página Web, que probablemente vaya a servir de índice entre los marcadores favoritos del usuario o incluso los buscadores lo utilizan para listar sus resultados de búsqueda. Un buen título de un sitio Web debe funcionar a modo de eslogan que al mismo tiempo describe en una breve frase el motivo de existencia de esa misma aplicación Web.

Así mismo, justo debajo de la barra principal de navegación y como antecedente del contenido de la página, siempre se sitúa un segundo título, marcado como tal mediante HTML, que informa al usuario en todo momento de cuál es la opción funcional de la aplicación en la que actualmente se encuentra.

Todos estos elementos ayudan a navegar y poner en situación al usuario de la aplicación Web, pero sin duda alguna el elemento de mayor relevancia para el usuario es el contenido de la propia página. Las barras de navegación o menús con enlaces son útiles y necesarios, pero nunca deben monopolizar la apariencia de la página Web ni dificultar la visibilidad de la información que ésta contiene. Por norma general, el contenido de una página Web debe ocupar al menos la mitad del espacio total de la pantalla, siendo el porcentaje ideal el 80%. En la aplicación Web que ocupa el presente proyecto, el contenido se presenta en el hueco natural generado por las barras de navegación, ocupando un amplio espacio y acercándose al porcentaje recomendado por los diseñadores Web.

La situación en la pantalla de los elementos anteriormente descritos puede apreciarse mediante la siguiente figura (Figura 6-4).

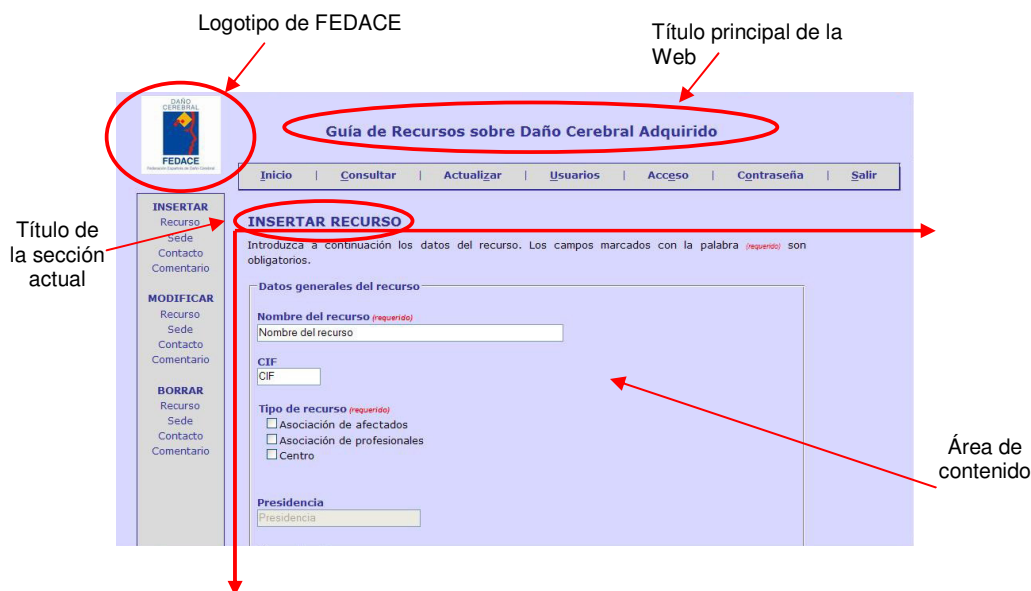


Figura 6-4 Otros elementos de la interfaz

Por último, mencionar que los elementos que se acaban de describir forman parte de todas las páginas que conforman la aplicación Web de gestión de la base de datos, con la única excepción de la página de identificación (Figura 6-5). Ésta proporciona información, centrada en la página, sobre el título principal de la aplicación Web, el logotipo de FEDACE y contiene además dos campos de texto editables, para introducir el nombre de usuario y la contraseña. De forma ocasional también dispondrá de un enlace de acceso directo a las consultas de recursos cuando el usuario “Administrador” haya establecido el modo de acceso libre en la aplicación.



Figura 6-5 Página de identificación

6.4.2 Guía de estilo empleada

La guía de estilo de un sitio Web representa una guía para todos aquellos elementos que dan un formato específico a sus contenidos, como por ejemplo el conjunto de colores, tipos de letra, el lenguaje utilizado, etc. En definitiva, es todo aquello que pretende transmitir al usuario una idea o imagen concreta sobre el sitio Web que está visitando.

Algunos de estos elementos han venido marcados desde el cliente y otros se han elegido libremente basándose en diferente bibliografía [Usabilidad: diseños de sitios Web, 2002], [Diseño de páginas Web, 1997]. Todos los aspectos de la guía de estilos se han marcado desde el uso de hojas de estilo o CSS, siguiendo el principio básico ya mencionado de separar siempre el contenido de la página de su aspecto y presentación.

6.4.2.1 Colores de la aplicación Web

El color es quizás uno de los elementos que mejor se transmite al usuario. Lo que se ha denominado como “psicología del color” es algo conocido desde hace mucho tiempo y está demostrado cómo puede influir éste en la forma de pensar, actuar y sentir de las personas. Por ejemplo, los colores muy luminosos o llamativos, como el rojo, el naranja o los amarillos anaranjados transmiten fuerza, dinamismo y vitalidad, pero también pueden provocar inquietud e incomodidad. Los colores pastel, sobre todo azules, verdes, tonalidades de rosa, blancos o grises inculcan una sensación de seguridad y de sosiego, pero también pueden llegar aburrir y a hacer pesado el contenido. Todo depende de los objetivos del sitio Web, el contexto de sus contenidos y la clase de usuarios a la que se pretende atraer.

A la hora de elegir los colores a utilizar en una Web, debe distribuirse por capas o grado de importancia. El color base y que dará uniformidad a la aplicación Web en desarrollo será el color elegido para el fondo. Lo más común es que el fondo sea el elemento más visible en toda la aplicación Web, por muchos contenidos, gráficos o barras de navegación que se coloquen por la pantalla. Dependiendo de este color se elegirán después los demás. Normalmente se busca que exista un fuerte contraste entre el color del fondo (preferiblemente blanco o color muy suave) con el color de los

elementos flotantes (menús, barras, logotipos, cuadros esquemáticos, etc.) y, sobre todo, con el texto, que de forma predominante suele ser negro.

La aplicación Web para la gestión de la base de datos tiene tres colores básicos y que le dan formato:

- Azul pastel para el fondo
- Gris para las barras de navegación
- Azul oscuro para los encabezados, los enlaces y el etiquetado de los formularios.

Estos colores fueron recomendados por el propio cliente, FEDACE. Manifestó desde el principio su deseo de guardar semejanza corporativa entre la actual Web de la que disponen y la futura Web de acceso a la base de datos. Ya que se iban a tratar de sitios Web diferentes, deseaban transmitir al menos la idea de que se tratan también de elementos complementarios y que versan sobre los mismos temas.

El color del fondo (azul pastel) y el de los elementos de navegación (gris) son colores muy claros y poco llamativos, que transmiten serenidad. Esta combinación de colores contrasta bastante bien con el color azul oscuro y el color negro elegidos para los encabezados, enlaces y el texto del contenido. Tal comprobación se realizó visionando la aplicación a través de una gama de grises gracias al software GrayBit [41] que permite visualizar cualquier Web proporcionada mediante su URL sin color y únicamente en blanco y negro (Figura 6-6).



Figura 6-6 Aplicación Web vista a través de escala de grises

Por otro lado, el color de los encabezados y el de los enlaces de hipertexto es el azul oscuro. Este color es fuertemente llamativo situado sobre los colores pastel del fondo y de las barras de navegación. Su visualización es también correcta si se basa en la escala de grises.

Por último, el color del texto es negro estándar. Es poco recomendable establecer un color diferente a éste para el texto, salvo casos muy excepcionales en los que el color del fondo recomiende otros. El color negro se lee mejor y permite además identificarlo ante el usuario como el contenido útil y real de la Web y no confundirlo con encabezados, comentarios, textos aclarativos u otros tipos de texto que por su naturaleza sí deben ser más llamativos. Para este último caso (mensajes de aviso o de error) se utiliza el color rojo, además del marcado correspondiente en HTML para subrayar el carácter extraordinario de estos mensajes.

6.4.2.2 Tipografía

El tipo de letra es factor importante puesto que determina la legibilidad de la lectura del contenido que, al fin y al cabo, será en lo que más se centre el usuario una vez localizada la información que estaba buscando. La legibilidad es un concepto que cambia en función del medio que se está utilizando: no es lo mismo la lectura sobre papel, como un libro o la prensa, que la lectura sobre una pantalla de ordenador.

El tipo de letra escogido para todos los elementos textuales de la aplicación Web (no sólo el contenido propiamente dicho, sino también vínculos y encabezados) es el tipo Verdana. Este tipo de letra está basado en otro, Sans Serif clásico, pero para un medio muy concreto: la pantalla. El espaciado entre letras, así como la anchura de sus caracteres, permite que se diferencien fácilmente en múltiples configuraciones de resolución, ayudando de esta forma a su lectura. Las curvas en las contraformas (perfil interno y externo de una letra) están sintetizadas al mínimo y las minúsculas son un píxel más altas que las contraformas de las mayúsculas en algunos cuerpos claves, para ayudar así a distinguir sin esfuerzo ciertos caracteres.

Por otro lado, el efecto de negrita se ha utilizado en el etiquetado de los campos en los formularios, en los enlaces de la barra principal de navegación y en los mensajes de

aviso como los de error, por ejemplo. Para los comentarios sobre los diferentes recursos se ha utilizado el efecto de cursiva o itálica, para diferenciarlos del resto de información sobre el propio recurso.

6.5 Mecanismos de seguridad

La necesidad de proporcionar mecanismos de seguridad en una aplicación Web vendrán determinadas por el tipo de contenido que ofrece, quién tiene autorización para acceder a ese contenido y aquellos componentes software que utiliza y que pueden comprometer no sólo al disponibilidad de la aplicación si no también las instalaciones informáticas sobre las cuales ésta se sustenta.

En este caso, la seguridad se orienta hacia dos puntos fundamentales: el control de accesos según el perfil del usuario y la consistencia de la base de datos.

6.5.1 Control de accesos

Dado que se trata de una aplicación Web orientada a diferentes usuarios que previamente deben ser dados de alta y que tienen, en función de su perfil, diferentes tipos de acceso, es fundamental diseñar el mecanismo mediante el cual este acceso va a ser controlado.

Los tipos de usuario, como se ha explicado anteriormente, pueden ser tres: “Administrador”, “Registrado” y “No Registrado”. De éstos, los dos primeros tipos son los que disponen de nombre de usuario y de clave de acceso. Es pues evidente la necesidad de establecer una página de identificación mediante un breve formulario donde se introduzcan y envíen estos datos.

La base de datos, además de contener todas las tablas descritas en su correspondiente capítulo, tiene además una tabla adicional, *USUARIOS*, que guarda los nombres de usuario, sus contraseñas y el tipo al que pertenecen (“Administrador” o “Registrado”). De esta forma, una vez rellenado el formulario de identificación y enviado éste al servidor, se comprueba si dicho nombre de usuario existe, y en tal caso, si la contraseña

introducida se corresponde con la almacenada. Una vez comprobados estos hechos, se comprueba además qué perfil es el que le corresponde al usuario. Todos estos datos se almacenan de forma local en el servidor como datos de sesión para posteriores comprobaciones.

Al mismo tiempo, en cada página de la aplicación Web existe una comprobación antes de llegar a procesarse ningún otro elemento de la página. Esta comprobación funciona planteando dos cuestiones:

- a) ¿El usuario se ha identificado?
- b) ¿El perfil del usuario permite acceder a esta página?

El motivo de incluir este sistema de seguridad adicional, además de la pantalla inicial de identificación, es para prevenir el hecho de que un usuario conozca por algún motivo la dirección URL exacta de una página interna de la aplicación Web e intente acceder a ella sencillamente escribiendo en el navegador la dirección en cuestión.

Todas estas comprobaciones se suman, además, a otras realizadas a lo largo de la propia página. Aunque un usuario esté identificado y además su perfil le permite acceder a la página a la que pretende llegar, es muy posible que el contenido que vaya a visualizar sea diferente al contenido visualizado por otro usuario con perfil distinto. Tal es el caso en la página de inicio o bienvenida, o en las barras de navegación que se reducen a una sola en caso de estar ante un perfil “Registrado”.

En el siguiente esquema (Figura 6-7) se puede apreciar una explicación gráfica de todo el proceso llevado a cabo en el control de accesos para los usuarios. En ella se muestran la página de identificación, el proceso de consulta a la base de datos, la verificación individual dentro de cada una de las páginas y la diferenciación del contenido de éstas en función de las variables de sesión que contienen el perfil del usuario.

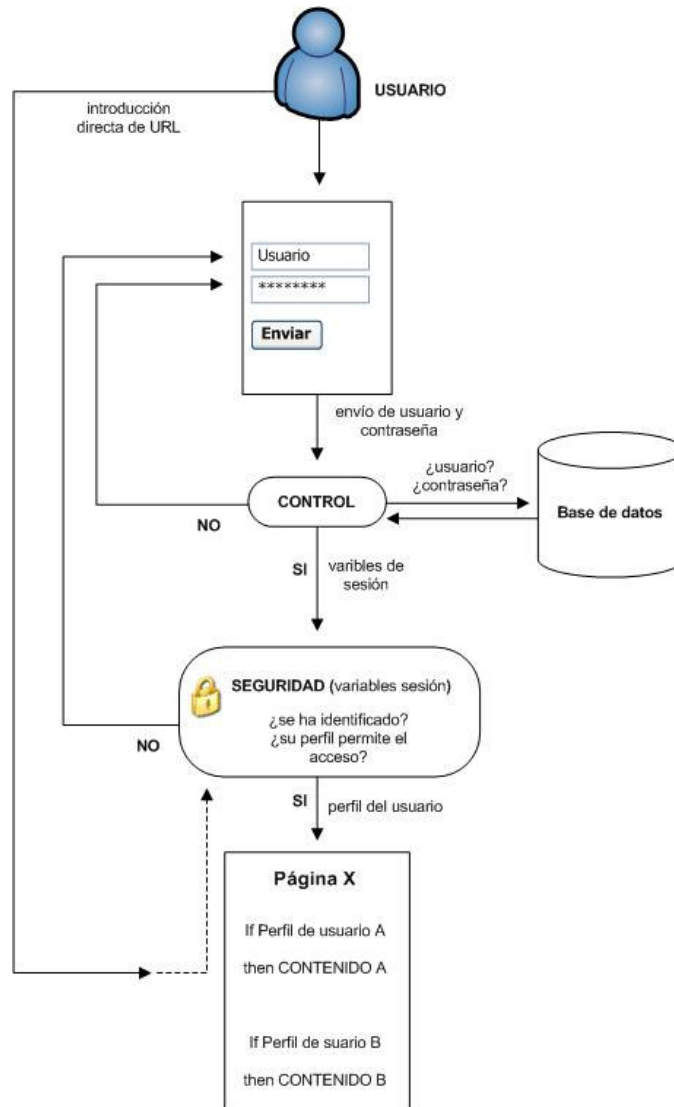


Figura 6-7 Control de accesos a la aplicación Web

6.5.2 Prevención de *SQL injection*

La conexión con la base de datos, que en el caso de la aplicación Web que ocupa este proyecto es algo constante, se realiza mediante sentencias de SQL para recuperar datos, insertarlos o editarlos. Estas sentencias se construyen utilizando en la mayoría de los casos unas plantillas estándar ya preparadas en el código de la página a las que simplemente se les añaden los datos a consultar o editar en los campos correspondientes de la sentencia. Los datos se añaden una vez que el usuario a rellenado el formulario correspondiente, lo ha enviado y los campos han sido comprobados y dados por válidos.

Por ejemplo, la sentencia de consulta formada para la identificación de un usuario que ya ha introducido y enviado sus datos puede ser, de forma estándar:

```
SELECT * FROM Usuarios WHERE nombre ='VARIABLE1' AND  
password='VARIABLE2';
```

Donde VARIABLE1 y VARIABLE2 serán los datos introducidos por el usuario. Sin embargo, pueden existir ciertos usuarios con intenciones maliciosas que conozcan esta típica forma de trabajar de las aplicaciones Web que conectan con bases de datos, y que aprovechen en su beneficio pequeños errores o desconocimientos por parte de los diseñadores de la misma. Una de las técnicas usadas por este tipo indeseable de usuarios es la conocida como *SQL injection*. Básicamente, consiste en intentar introducir de forma premeditada determinadas sentencias SQL para que éstas sean lanzadas contra la base de datos, permitiendo así desde un acceso no autorizado a determinadas áreas de la Web hasta la modificación o borrado completo de la base de datos.

Por ejemplo, un caso típico para esta técnica es el habitual formulario de identificación donde se nos pide nombre de usuario y contraseña. Si, por ejemplo, se introduce el nombre de usuario “usuarioA” y la contraseña “miPass”, la sentencia SQL que la aplicación construirá para comprobar si existe será algo similar a esta

```
SELECT * FROM Usuarios WHERE nombre ='usuarioA' AND password='miPass';
```

Sin embargo, un usuario malicioso y conocedor de las técnicas de inyección de SQL podría insertar como usuario “usuarioA” y como contraseña “ 'OR 1=1 -- ”. El resultado sería la creación de la siguiente sentencia SQL:

```
SELECT * FROM Usuarios WHERE nombre ='usuarioA' AND password=''OR 1=1-  
-';
```

Esta sentencia, como se puede comprobar, se ejecutaría sin problema alguno puesto que aunque el usuario “usuarioA” no exista y el campo “password” esté comparado con la cadena vacía, el valor 1 siempre es igual 1. El usuario malicioso acabaría de conseguir el acceso al sistema, y éste no es más que un ejemplo de las múltiples posibilidades que se pueden intentar con el uso de esta técnica.

La prevención de este problema puede hacerse aplicando una serie de recomendaciones básicas, que son las que se han puesto en práctica durante el diseño de la aplicación Web de gestión de la base de datos:

- Escapar todas las cadenas y valores introducidos por el usuario a través de formularios o similares. Escapando una cadena se anulan los caracteres especialmente sensibles en la construcción de sentencias, como la comilla doble “”, la comilla simple ‘’, la barra inclinada / y otros caracteres.
- Nunca pasar a través de la URL variables que contengan sentencias SQL. Es hasta cierto punto habitual, encontrar páginas que redirección a otras pasándoles la sentencia SQL a ejecutar que pueden ser fácilmente modificadas.
- No mostrar los mensajes de error que por defecto generan las aplicaciones, sobre todo las de bases de datos. Un mensaje de error puede dar valiosas pistas sobre cuál es la estructura de la base de datos, sus tablas y los atributos de éstas.
- No conectar a la base de datos mediante el perfil de administrador, a no ser que se estrictamente necesario. Es mejor definir una serie de perfiles con sus correspondientes permisos.
- Limitar la concatenación de múltiples sentencias de ejecución. Lo ideal es lanzar únicamente una sentencia por cada conexión con la base de datos.

7 DISEÑO ACCESIBLE DE LA APLICACIÓN WEB

Uno de los aspectos más importantes y relevantes del diseño de la aplicación Web de gestión de la base de datos de FEDACE es la accesibilidad. Desde el primer momento se mostró por parte de FEDACE un gran interés en lograr que la futura aplicación Web fuera plenamente accesible, ya que las circunstancias personales de los usuarios y el cometido de la ONG así lo exigían.

Como ya se ha dicho en capítulos anteriores, la accesibilidad implica un acceso universal a la Web, no teniendo que depender del software o hardware utilizado, del idioma o la cultura del usuario o creador Web, de la infraestructura de red o de las capacidades físicas de los usuarios. Un diseño orientado a la accesibilidad no solamente está beneficiando a usuarios con alguna discapacidad, sino que también beneficia al resto de usuarios en su interacción con el material Web.

La accesibilidad hasta ahora no había tenido demasiada repercusión en los diseñadores de sitios Web y en muchas ocasiones se aplicaban técnicas erróneas o conceptos equivocados. Hoy día, sin embargo, parece que está surgiendo más interés en este área y mayor afán por parte de los diseñadores de obtener no sólo una Web atractiva, usable y bien estructurada, sino también una Web que cumpla con la normativa vigente y los estándares existentes sobre accesibilidad.

Desde el W3C, organismo internacional encargado de realizar estándares y protocolos de diseño Web, se lanzó en 1999 un documento con rango de recomendación (el más importante de sus trabajos) acerca de las normas y pasos para llegar a un diseño Web accesible: las “Pautas de Accesibilidad al Contenido en la Web 1.0” [27]. Así mismo, se creó un grupo de trabajo y divulgación específico para esta materia denominado WAI [3].

Las “Pautas de Accesibilidad al Contenido en la Web 1.0” tienen además una serie de documentos asociados que abordan de forma más concreta los diferentes aspectos que abarca un diseño accesible. Así, también están disponibles las “Técnicas Fundamentales para las Pautas de Accesibilidad al Contenido en la Web 1.0” [38], las “Técnicas CSS

para las Pautas de Accesibilidad al Contenido en la Web 1.0” [39] y las “Técnicas HTML para las Pautas de Accesibilidad al Contenido de la Web 1.0” [40].

Los citados documentos, todos ellos disponibles a través del W3C, han sido la guía de referencia y consulta en la elaboración de este diseño accesible para la aplicación Web de gestión de la base de datos.

7.1 Introducción a las pautas para un diseño accesible

Los creadores de páginas y portales Web siempre analizan y estudian el tipo de usuarios al que deben orientar el diseño para hacerla más atractiva y con éxito. Puede pensarse, por tanto, que el diseño accesible sólo debería tenerse en cuenta en aquellos casos en los que se conoce de antemano que los futuros usuarios presentan algún tipo de discapacidad. Nada más lejos de la realidad, más aún cuando la Web se ha convertido en un medio de masas donde el usuario potencial puede ser prácticamente cualquiera con conexión a Internet. Todo diseñador debería tener en cuenta el amplísimo abanico de posibles circunstancias bajo las cuales, ese usuario potencial que previamente han identificado, puede conectarse a su Web.

Por ejemplo, no todos los usuarios no discapacitados tienen por qué tener la misma capacidad de lectura comprensiva, nivel de audición o agudeza visual. No será infrecuente, debido al salto dado por Internet desde el ordenador personal de sobremesa a todo tipo de dispositivos móviles, que la pantalla en la cual visualizan los contenidos presente solamente texto, sea excesivamente pequeña o la conexión sea más lenta o inconstante de lo normal. El usuario no tiene por qué hablar y leer con fluidez el idioma en el que la Web está escrita. Con respecto al navegador utilizado, es muy posible que no sea el navegador más frecuentemente usado por una amplia comunidad de usuario o la última versión lanzada del mismo. Por último, el contexto bajo el cual navega un usuario es del todo desconocido: puede estar ocupando su vista, oídos o manos en cualquier otra tarea diferente a la navegación por la Web, como puede ser conducir o trabajar en un entorno ruidoso, por poner unos ejemplos.

Las pautas abordan los diferentes aspectos de la accesibilidad Web y proporcionan soluciones de diseño accesibles. Señalan aquellas típicas situaciones que pueden suponer un problema a los usuarios que presentan alguna discapacidad. Para ello, las pautas se basan en dos motivos principales: guiar a una transformación del contenido correcta y conseguir que este contenido sea comprensible y navegable.

En total, el documento “Pautas de Accesibilidad al Contenido en la Web 1.0”, contiene 14 pautas o principios fundamentales en el diseño accesible. A su vez, las pautas definen una serie de puntos de verificación a través de los cuales se puede comprobar el nivel de cumplimiento de la pauta a la que pertenecen.

En función de su importancia, los puntos de verificación de cada pauta de accesibilidad se clasifica según tres prioridades:

- Prioridad 1: un desarrollador de contenidos de páginas Web **tiene** que satisfacer este punto de verificador. De otra forma, uno o más grupos de usuarios encontrarán imposible acceder a la información del documento. Satisfacer este punto de verificación es un requerimiento básico para que algunos grupos puedan usar los documentos Web [38].
- Prioridad 2: un desarrollador de contenidos de páginas Web **debe** satisfacer este punto de verificación. De otra forma, uno o más grupos encontrarán dificultades en el acceso a la información del documento. Satisfacer este punto de verificación eliminará barreras de acceso en los documentos Web [38].
- Prioridad 3: un desarrollador de contenidos de páginas Web **puede** satisfacer este punto de verificación. De otra forma, uno o más grupos de usuarios encontrarán alguna dificultad para acceder a la información del documento. Satisfacer este punto de verificación mejorará la accesibilidad de los documentos Web [38].

Como conclusión, en el nivel de accesibilidad de una Web vendrá determinada por el número de puntos de verificación y su prioridad que cumpla:

- Nivel A: se satisfacen todos los puntos de verificación del prioridad 1.
- Nivel AA: se satisfacen todos los puntos de verificación de prioridad 1 y 2
- Nivel AAA: se satisfacen todos los puntos de verificación de prioridad 1, 2 y 3.

7.2 Pautas de Accesibilidad al Contenido en la Web

A continuación se enumeran las catorce pautas existentes en el documento “Pautas de Accesibilidad al Contenido en la Web 1.0” [38]. Para cada una de ellas, se expone el diseño propuesto para cubrir cada uno de sus puntos de verificación con el objetivo principal de llegar a obtener el nivel de conformidad AAA. También se exponen ejemplos del código HTML o CSS a modo de ejemplo o acompañamiento a la pauta y punto de verificación correspondiente. Este código en ocasiones no mostrará todas sus características reales presentes en el código fuente, sino que se centrará en aquellos aspectos más relevantes para el ejemplo.

En algunas de estas pautas ciertos puntos de verificación, o a veces su totalidad, no pueden ser aplicadas a la aplicación Web de gestión de la base de datos que ocupa el presente proyecto, puesto que ésta carece de las funcionalidades o elementos Web de los que en esas pautas o puntos de verificación se habla.

7.2.1 Pauta 1

“Proporcione alternativas equivalentes para el contenido visual y auditivo.”

Esta pauta explica la necesidad de ofrecer un contenido que cumpla en esencia la misma función que los elementos visuales o auditivos proporcionados en la Web. En este tema es fundamental aplicar lo que se denomina equivalencia textual o descripción textual del texto, que puede ser interpretada mediante braille, sintetizadores de voz o como texto visible por aquellos dispositivos de navegación utilizados por personas con alguna discapacidad.

Puntos de verificación aplicables en esta pauta:

1.2 *“Proporcione un texto equivalente para todo elemento no textual (por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). Esto incluye: imágenes, representaciones gráficas del texto (incluyendo símbolos), mapas de imagen, animaciones (por ejemplo, GIFs animados), applets y objetos programados, ASCII art, marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones*

gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos.” (Prioridad 1)

En el presente caso la Web contiene imágenes, pero no gráficos, mapas de imagen, botones creados con gráficos, ni ningún tipo de animación, vídeos o sonidos. Las imágenes presentes en la aplicación Web se corresponden con el logotipo de FEDACE en la esquina superior izquierda y los iconos de validación de accesibilidad, HTML y CSS en la parte inferior de las páginas de la aplicación Web.

Mediante el atributo ALT se puede describir de forma breve el contenido de la imagen o su descripción, de forma que una persona que no pueda visualizar la imagen, leyendo o escuchando por un sintetizador este texto pueda comprender qué es lo que simboliza o representa. El atributo LONGDESC permite hacer descripciones más exactas y extendidas sobre la imagen, en caso de que su nivel de detalle sea considerable y sea necesaria una explicación larga para comprender su significado.

El atributo ALT fue el escogido para describir las imágenes presentes en la página, que son, respectivamente los logotipos de FEDACE, nivel de conformidad AAA del W3C, HTML 4.01 válido del W3C y CSS válida del W3C:

```

```

```

```

```

```

```

```

7.2.2 Pauta 2

“No se base sólo en el color.”

La segunda pauta previene el hecho de que se use un color determinado en el texto o en zonas específicas de la Web como medio transmisor de alguna información. Por ejemplo, el usuario un color muy llamativo para destacar noticias de última hora o mensajes importantes, así como indicaciones del tipo “Por favor, seleccione una opción de entre las de color rojo”. Esto dificulta la comprensión del contenido y la interacción a aquellas personas que no utilicen navegadores a color o que tenga una discapacidad visual que les impida diferenciar los contrastes entre colores.

Los puntos de verificación aplicables son:

2.1 “Asegúrese de que toda la información transmitida a través de los colores también esté disponible sin color, por ejemplo mediante el contexto o por marcadores.”
(Prioridad 1)

Por ejemplo, todos los títulos dentro de la aplicación Web tienen un color más oscuro y tamaño de letra mayor y más ancho en función de su importancia. Para hacerlos accesibles y seguir un correcto marcado, los títulos se han etiquetado con los encabezados correspondientes desde el H1 hasta el H3.

```
<h1>Guía de Recursos sobre Daño Cerebral Adquirido</h1>
...
<h2>CONSULTAR RECURSO</h2>
...
<h3>Federación Española de Daño Cerebral</h3>
...
<h3>Centro Hospitalario Benito Menni</h3>
...
<h3>Unidad de Daño Cerebral Casaverde </h3>
...
```

De la misma forma, existen mensajes de advertencia o resaltados en color rojo para, por ejemplo, señalar que un campo de un formulario es obligatorio. Estas indicaciones,

además de ir en color rojo, están escritas en un tipo de letra y tamaño distintos y etiquetadas con el elemento EM de énfasis:

```
<legend>
  <span class="remarcado">Tipo de recurso
  <em><span class="obligatorio"> (requerido)</span></em>
  </span>
</legend>
```

2.2 “*Asegúrese de que las combinaciones de los colores de fondo y primer plano tengan suficiente contraste para que sean percibidas por personas con deficiencias de percepción de color o en pantallas en blanco y negro.*” (Prioridad 2 para las imágenes, Prioridad 3 para el resto)

Los colores básicos usados en la aplicación Web son un fondo de azul claro, diversas tonalidades de grises para las barras de navegación y un azul más oscuro para encabezados y títulos. El texto conserva un color oscuro. Esta elección no fue tomada por el desarrollador sino que respondía al estilo corporativo usado por FEDACE en su actual Web de presentación.

Para asegurar el cumplimiento de este punto de verificación, se utilizó una herramienta llamada GrayBit [41] para transformar el aspecto completo de la aplicación Web en una versión basada en escala de grises. De esta forma se experimenta la misma situación que aquellas personas que para navegar por la Web utilizan monitores en blanco y negro o tienen deshabilitado el color. La prueba permitió concluir que en contraste entre los colores del fondo, de las barras de navegación y del texto de la aplicación Web era el suficiente como para no representar problemas de visionado¹.

Así mismo, se utilizan hojas de estilo donde toda la gama de colores queda definida. Los colores se identifican por códigos numéricos o porcentajes y no por nombres. Así mismo, se hace uso de las propiedades respectivos de CSS en la configuración del color como *color*, *background-color* o *border-color*. Por ejemplo:

¹ Consultar capítulo 6, apartado sobre *Colores de la aplicación Web*.

```
A.menu:hover{  
    color: rgb(255,255,255);  
    text-decoration: none;  
    font-weight: bold;  
    background-color: rgb(51,51,153);  
}
```

7.2.3 Pauta 3

“Utilice marcadores y hojas de estilo y hágalo apropiadamente.”

Es muy importante seguir el principio básico en la construcción de páginas Web de diferenciar claramente la estructura del documento de su presentación. HTML es un lenguaje de marcado de hipertexto, es decir, especifica la estructura del texto y lo divide en partes lógicas y diferenciadas. Un error muy frecuente hasta ahora era el de utilizar algunos elementos HTML para crear efectos en las fuentes, colores, tamaños o en general, cualquier característica del maquetado de la Web. Este tipo de prácticas dificulta enormemente la accesibilidad de todos los posibles usuarios.

Algunos de los puntos de verificación aplicables son:

3.2 “*Cree documentos que estén validados por las gramáticas formales publicadas.*” (Prioridad 2)

Todos los documentos HTML asociados a la aplicación Web han sido validados mediante los validadores automáticos disponibles en el W3C [42], que han certificado que el HTML usado se ajusta de forma estricta a la gramática correspondiente a HTML 4.01 *Transitional* [45]. De igual forma, las hojas de estilo han sido correctamente validadas con este mismo sistema, dando una gramática perfectamente ajustada a los estándares actuales.

Para ambos casos, se han colocado en las páginas Web sendos logotipos que informan de la conformidad de las páginas con las gramáticas formales del W3C.

3.3 “Utilice hojas de estilo para controlar la maquetación y la presentación.” (Prioridad 2)

Como se ha mencionado anteriormente, es sumamente importante separar claramente la estructura de la Web de su formato. El lenguaje HTML debe utilizarse únicamente para marcar las diferentes partes del documento y nunca para darle una apariencia concreta.

Para dar el formato y aspecto deseado al documento se deben utilizar las hojas de estilo o CSS. Mediante sus propiedades, se puede alterar prácticamente cualquier elemento de la Web, cambiando el color, el tamaño, la fuente o la posición, entre otros. Por ejemplo, mediante el uso de las propiedades *text-indent*, *text-align*, *word-spacing* y *font-stretch* se puede configurar el espaciado entre caracteres y palabras, el aspecto de la primera línea de un texto, entre otros muchos. Se recomienda no utilizar tablas o elementos HTML similares para conseguir la colocación exacta de algún texto u otro elemento dentro del documento, sino se debe procurar utilizar propiedades CSS para su posicionamiento. Las propiedades *float*, *position*, *top*, *right*, *bottom* y *left* representan en la mayoría de los casos una solución idónea para situar en cualquier punto del documento el texto deseado, sin tener que recurrir a elementos HTML.

Por ejemplo, para situar el logotipo de FEDACE se utiliza el siguiente fragmento de CSS:

```
div.imagen {  
    top:0.5em;  
    left: 2.5em;  
    height: 3em;  
    width: 9em;  
    position: absolute;  
    visibility: visible;  
}
```

Otro ejemplo, anteriormente visto, se utiliza el elemento HTML de énfasis, EM, para indicar correctamente la importancia de una parte del texto (en este caso, indicar un elemento de formulario obligatorio) en vez de utilizar otros elementos HTML, como la negrita B o la cursiva I, para darle un formato llamativo pero que estructuralmente

estaría mal utilizado. Obsérvese que también se hace uso de propiedades de CSS para darle el aspecto deseado, en este caso en negrita (el título del campo) y de color rojo (la indicación de obligatoriedad):

```
<span class="remarcado">Nombre del recurso
<em><span class="obligatorio"> (requerido)</span></em>
</span>
<input type="text" name="nombrerecurso" value="Nombre del recurso"
size="60" maxlength="100">
```

3.4 “Utilice unidades relativas en lugar de absolutas al especificar los valores en los atributos de los marcadores de lenguaje y en los valores de las propiedades de las hojas de estilo.” (Prioridad 2)

Se deben utilizar unidades de medida relativas en las hojas de estilo, como pueden ser *em* y porcentajes, frente a otras unidades absolutas que son menos accesibles, como los puntos, *pt*, o los centímetros, *cm*. Las medidas absolutas únicamente deberían ser usadas cuando se conoce sin lugar a dudas las características del dispositivo de visionado. De otra forma, al cambiar el tamaño de las fuentes o la resolución de pantalla original, la colocación de los elementos pueden ser totalmente diferente a la deseada por el desarrollador, empobreciendo el aspecto de la Web y llegando a impedir la comprensión de la misma.

En las hojas de estilo de la presente Web, todas las unidades de medida son relativas y se basan principalmente en la unidad *em*, por ejemplo:

```
span.submenu {
    font-size:0.9em;
    padding-left: 0.5em;
    padding-right: 0.5em;
}
```

3.5 “Utilice elementos de encabezado para transmitir la estructura lógica y utilícelos de acuerdo con la especificación.” (Prioridad 2)

Los elementos de encabezado de HTML son desde H1, el nivel más importante, al H6, el encabezado más simple. Por ejemplo, no se debería utilizar nunca un nivel H3 porque su tamaño se ajuste al aspecto que se busca, si no se ha utilizado anteriormente un nivel H2 para indicar una sección que engloba a la anterior. Igualmente, si se está buscando indicar un título o un apartado, no se debería utilizar una fuente modificada sin más mediante CSS con un tamaño más grande o color distinto, sino que se debería usar un elemento de encabezado correspondiente y modificar, mediante propiedades CSS, el aspecto que se esté buscando.

En la aplicación Web de gestión de la base de datos de FEDACE se han utilizado encabezados hasta de nivel 4, como ya se ha indicado en puntos de verificación anteriores, respetando siempre el orden y nivel que les corresponde en cada caso. Para ajustar su tamaño y color se utilizan siempre hojas de estilo, modificando así las propiedades que por defecto puedan tener los elementos de encabezado en HTML.

3.6 “Marque correctamente las listas y los ítems de las listas.” (Prioridad 2)

Para aquellos elementos del texto que constituyan una enumeración o una estructura que podría entenderse como una lista de elementos deben utilizarse las etiquetas HTML correspondientes para ello. Las listas facilitan la comprensión y navegación a aquellos usuarios con discapacidad visual. Nunca sería correcto utilizar el elemento DL, UL y OL de HTML para listas con el objetivo que conseguir sangrías o efectos concretos en el texto. Para ello están las hojas de estilo.

Por ejemplo, alguna de las listas utilizadas en la aplicación Web es la usada para enumerar los servicios ofrecidos por un recurso:

```
<span class='simple'>Servicios:</span>
<ul>
  <li>Fisioterapia
  <li>Logopedia
  <li>Medicina general
  <li>Neurología
```

```
<li>Neuropsicología
<li>Terapia ocupacional
</ul>
```

7.2.4 Pauta 4

“Identifique el idioma usado.”

La identificación del idioma en el documento permite a aquellos usuarios que utilizan dispositivos especiales de navegación comprender y utilizar de forma más completa la aplicación Web. Los sintetizadores de voz o los componentes braille son capaces de adaptarse al idioma utilizado en la Web haciendo más accesible el documento y evitando así que se vuelva completamente indescifrable. La especificación del idioma también ayuda al resto de usuarios a utilizar cualquier software de traducción automática.

Los puntos de verificación aplicados en esta Web son los siguientes:

4.1 “Identifique claramente los cambios en el idioma del texto del documento y en cualquier texto equivalente (Por ejemplo, leyendas).” (Prioridad 1)

El idioma principal de la aplicación Web es el castellano, pero dado que la base de datos contempla los recursos existentes en toda España, a veces pueden aparecer elementos que están en otras lenguas españolas como catalán, vasco o gallego.

Es el caso, por ejemplo, del los tipos de vía existentes para especificar la dirección completa de un recurso. Muchas denominaciones de vías tienen, en su respectiva comunidad autónoma, su nombre oficial en la lengua cooficial al castellano. En tal caso, se especifica el idioma mediante el atributo LANG, siendo la abreviatura “ca” la oficial correspondiente al catalán, “eu” la correspondiente al vasco y “gl” la usada para el gallego:

Tipo de vía

```
<select name="TipoVia" tabindex="57">
  <option value="Avenida">Avenida</option>
  <option value="C/">Calle</option>
  <option value="Carrer" lang="ca">Carrer</option>
```



```
<option value="Glorieta">Glorieta</option>
<option value="Kalea" lang="eu">Kalea</option>
<option value="Paseo">Paseo</option>
<option value="Passeig" lang="ca">Passeig</option>
<option value="Plaça" lang="ca">Plaça</option>
<option value="Plaza">Plaza</option>
</select>
```

4.2 “Especifique la expansión de cada abreviatura o acrónimo cuando aparezcan por primera vez en el documento.” (Prioridad 3)

Las abreviaturas se indican con el elemento HTML destinado para ello, <ABB>. Mediante el atributo TITLE, debe indicarse el significado completo de la abreviatura que se está señalando la primera vez que ésta aparece.

Por ejemplo, a la hora de indicar una dirección se utilizan la siguientes abreviaturas:

```
<abbr title="Calle">C/ </abbr> Alameda Apodaca nº 12, Bajo izq
<abbr title="Código Postal">C.P: </abbr> 11003
```

4.3 “Identifique el idioma principal de un documento.” (Prioridad 3)

En todos y cada uno de los documentos HTML utilizados en la elaboración de la aplicación Web de gestión de la base de datos se especifica claramente que el idioma utilizado es el español o castellano. A pesar de ello, y como se ha explicado en puntos de verificación anteriores, en caso de existir un cambio de idioma se especifica claramente.

La cabecera de todos los documentos HTML es:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="es">
```

7.2.5 Pauta 5

“Cree tablas que se transformen correctamente.”

Las tablas en HTML están pensadas para presentar aquella información que tiene pleno significado sólo si se muestra de forma tabular. Los elementos destinados a crear tablas nunca deberían usarse para maquetar el contenido de la Web o colocarlo en una posición o formato concreto, a no ser que sea bajo una condiciones muy concretas. Las tablas correctamente definidas en HTML permiten a los dispositivos de navegación usados por las personas con alguna discapacidad ofrecer e interpretar el contenido de forma clara y comprensible. Así mismo, las tablas usadas de manera correcta facilitan la lectura y la navegación por la Web a todo tipo de usuarios.

Los puntos de verificación de esta pauta que se pueden aplicar a la presente Web son los siguientes:

5.1 “En las tablas de datos, identifique los encabezamientos de fila y columna.” (Prioridad 1)

Es importante no sólo mostrar los datos de forma tabular cuando su completa comprensión lo requiera si no además tabularlos de forma correcta. Para ello, los encabezados de columna o fila permiten una mayor facilidad de lectura al usuario y además ayuda a los dispositivos de navegación para personas con discapacidad.

Para realizar un encabezado, se hace uso de los atributos ID, HEADERS y el elemento TH, que permiten indicar el encabezado concreto de la tabla. A continuación se muestra un ejemplo con la tabla dedicada a mostrar el equipo de un recurso, con todos los posibles cargos y el número de ellos (se muestra sólo un fragmento del código):

```
<table class="equipo" id="equipo">
<tr>
  <th id="header1">Cargo</th>
  <th id="header2">Nº</th>
  <th id="header3">Cargo</th>
  <th id="header4">Nº</th>
</tr>
```

```

<tr>
<td class="equipo" headers="header1">
    <input type="checkbox" name="equipo1" value="Administrativo">
    Administrativo
</td>
<td headers="header2">
    <input type="text" name="numero1">
</td>

<td class="equipo" headers="header3">
    <input type="checkbox" name="equipo2" value="Monitor ocio y
    tiempo libre">
    Monitor ocio y tiempo libre
</td>

<td headers="header4">
    <input type="text" id="numequipo2" name="numero2">
</td>
</tr>

<tr>
<td class="equipo" headers="header1">
    <input type="checkbox" name="equipo3" value="Animador socio-
    cultural" >
    Animador socio-cultural
</td>

<td headers="header2">
    <input type="text" name="numero3">
</td>

<td class="equipo" headers="header3">
    <input type="checkbox" name="equipo4" value="Neurocirujano">
    Neurocirujano
</td>

<td headers="header4">
    <input type="text" name="numero4">
</td>
</tr>

```

El aspecto de este fragmento de tabla es algo similar a lo siguiente (Figura 7-1):

Cargo	Nº	Cargo	Nº
<input type="checkbox"/> Administrativo	<input type="checkbox"/> nº	<input type="checkbox"/> Monitor ocio y tiempo libre	<input type="checkbox"/> nº
<input type="checkbox"/> Animador socio-cultural	<input type="checkbox"/> nº	<input type="checkbox"/> Neurocirujano	<input type="checkbox"/> nº

Figura 7-1 Tabla en HTML

5.3 “No utilice tablas para maquetar, a menos que la tabla tenga sentido cuando se alinee. Por otro lado, si la tabla no tiene sentido, proporcione una alternativa equivalente (la cual debe ser una versión alineada).” (Prioridad 2)

Las tablas no deben usarse para posicionar elementos en el lugar deseado, sino para presentar datos que tienen significado de forma tabular. En caso de usar una tabla para presentar elementos que no necesariamente precisan de ésta, es necesario que los datos una vez alineados conserven su sentido.

En la aplicación Web de acceso a la base de datos, a la hora de mostrar un formulario con los campos “Servicio” y “Modelo de atención” se ha optado por usar una tabla, sin encabezamiento ni títulos, para evitar la prolongación excesiva de la longitud de la página. En este caso, los datos una vez alineados conservan perfectamente el significado que pretenden ofrecer, sin dificultar la navegación por ellos mediante dispositivos especiales para personas con alguna discapacidad.

5.4 “Si se utiliza una tabla para maquetar, no utilice marcadores estructurales para realizar un efecto visual de formato.” (Prioridad 2)

En las tablas utilizadas para los campos “Servicios” y “Modelo de atención”, en aquellos formularios de la aplicación Web donde aparecen, no se han utilizado ninguno de los elementos HTML o atributos de encabezamiento de tablas (ID, HEADERS, TH) para no confundir a los dispositivos de navegación y evitar la idea de que se tratan de datos presentados de forma tabular porque lo requiera su completa comprensión.

5.5 “Proporcione resúmenes de las tablas.” (Prioridad 3)

Los resúmenes de tablas son especialmente útiles para aquellos usuarios invidentes que utilicen navegadores basados en sintetizadores de voz o similares. No sólo se debe describir el objetivo de la tabla o la naturaleza y origen de los elementos contenidos en ella sino también una explicación sobre la relación entre la tabla y el contexto general del documento en el que se enmarca.

Mediante el atributo SUMMARY del elemento TABLE se puede proporcionar un resumen general de la tabla. Para la tabla “Equipo”, que enmarca los cargos y número de empleados con ese cargo dentro de un recurso, se define la tabla de la siguiente manera:

```
<legend class="g">
```

```
<span class="remarcado">Equipo</span></legend>
```

```
<table class="equipo" id="equipo" border="1" summary="Tabla con todos  
los posibles profesionales con los que cuenta la sede del recurso y el  
número de ellos que hay disponibles (opcional)">
```

7.2.6 Pauta 6

“Asegúrese de que las páginas que incorporan nuevas tecnologías se transformen correctamente.”

Cada vez es más frecuente encontrar sitios Web con infinidad de elementos relacionados con lo último en tecnología de vídeo, audio y animaciones de todo tipo. Estos efectos hacen más atractiva la Web y ayudan, en algunos casos, a interactuar mejor con los contenidos y a mostrar éstos de una forma más clara y concisa. Sin embargo, la otra cara son los inconvenientes en cuanto a accesibilidad que pueden llegar a presentar estos elementos.

Como principio básico, se debe intentar que la Web sea capaz de ofrecer el mismo contenido y con las mismas posibilidades de comprensión por parte de los usuarios cuando éstos no tienen habilitadas o disponibles las nuevas tecnologías empleadas.

Aquellos puntos de verificación de esta pauta que se pueden aplicar a la aplicación Web de gestión de la base de datos son los que figuran a continuación:

6.1 *“Organice el documento de forma que pueda ser leído sin hoja de estilo. Por ejemplo, cuando un documento HTML es interpretado sin asociarlo a una hoja de estilo, tiene que ser posible leerlo.”* (Prioridad 1)

Como se ha mencionado anteriormente, es fundamental en el desarrollo de una Web diferenciar claramente su estructura, marcada por el etiquetado en HTML, de su presentación, obtenida gracias a las hojas de estilo o CSS. Mediante las hojas de estilo se puede dar cualquier apariencia deseada a todos los elementos que componen una Web. Incluso permiten la posibilidad de situar en la Web los contenidos en un orden diferente al que están declarados en el etiquetado mediante HTML.

Sin embargo, en muchas ocasiones, los navegadores usados por los usuarios, en especial si presentan alguna discapacidad, puede que no sean capaces de ofrecer todos los formatos y apariencias dadas al documento a través de las hojas de estilo. Puede darse el caso de que el documento se muestre casi en exclusiva a través del etiquetado dado con el lenguaje HTML. En tal caso, es muy importante que el documento esté organizado de forma lógica para que la navegación por él, en caso de que no tenga el formato dado o las hojas de estilo estén deshabilitadas, sea clara y evidente para el usuario.

La aplicación Web de acceso a la base de datos está construida de tal forma que su contenido mantiene un orden lógico e intuitivo de navegación aún en el caso de que las hojas de estilo que le dan formato se deshabiliten por completo. Cuando esto sucede, los elementos aparecen de la siguiente forma (Figura 7-2):

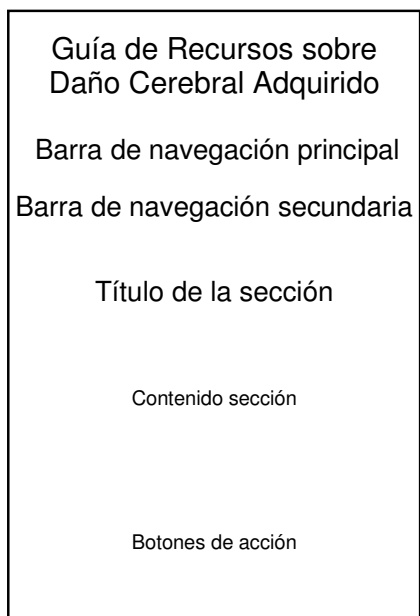


Figura 7-2 Apariencia de la página Web sin hoja de estilos habilitada

6.3 “Asegúrese de que las páginas sigan siendo utilizables cuando se desconecten o no se soporten los scripts, applets u otros objetos programados. Si esto no es posible, proporcione información equivalente en una página alternativa accesible.” (Prioridad 1)

Los *scripts* o cualquier contenido dinámico dentro de una página Web han de ser plenamente accesible, lo que significa que aquellas personas que no puedan utilizar estos elementos deben poder disponer de la misma funcionalidad o servicio que proporcionan de forma alternativa. Estos casos pueden darse de forma habitual en los navegadores adaptados a personas con alguna discapacidad o en los navegadores del resto de usuarios si no son una versión compatible con estos elementos dinámicos.

Los *scripts* escritos en Javascript son una herramienta frecuente en la aplicación Web de acceso a la base de datos, principalmente para la validación de formularios y para la actualización automática de menús de selección. Sin embargo, se buscó en todo momento que la función que realizaban estos scripts estuviera igualmente disponible en caso de que el navegador utilizado por el usuario no pudiera ejecutar este tipo de elementos. Para ello se han utilizado, principalmente, los elementos SCRIPT y

NOSCRIPT de HTML. A continuación se muestran las soluciones encontradas en los principales casos de uso de *scripts*.

➤ Validación de formularios.

La validación de los campos que constituyen los distintos formularios presentes en la aplicación Web se realizan mediante dos vías: por Javascript y con PHP. Al enviar un formulario, pulsando el botón de envío correspondiente, de forma automática se lanzan una serie de funciones implementadas en Javascript que comprueban cada uno de los campos del formulario y avisan de posibles errores. Sin embargo, puede ocurrir que el navegador del usuario no soporte este formato. Para tales casos, se ha utilizado el lenguaje del lado del servidor PHP para conseguir igualmente una validación de datos. Ésta se realiza en el servidor Web, por lo que es necesario enviar previamente los datos para poder comprobarlos, al contrario de lo que sucede con Javascript donde es el propio cliente el que ejecuta las comprobaciones.

Todas las funcionalidades de la aplicación Web están implementadas siguiendo un esquema básico de funcionamiento: una página que muestra el formulario de inserción de datos, y otra que los recoge e interactúa con ellos y la base de datos, según la operación a realizar. En el caso de funcionar Javascript, las comprobaciones se lanzarían en la primera página, nada más pulsar el botón de “Enviar”. Sin embargo, en la segunda página que es la que recopila los datos introducidos, todos y cada uno de los campos son comprobados nuevamente mediante PHP. Si Javascript estuviera deshabilitado o no funcionase, la comprobación de los datos se realizaría una única vez mediante PHP y en la segunda página, la de recogida de datos.

Los botones de envío presentes en las páginas con formulario siguen un esquema parecido al siguiente:

```
<script type="text/javascript">
    document.write("<input type='button' value='INSERTAR COMENTARIO'
        onClick='validarComentario(this.form) '>")
</script>
<noscript>
    <input type="submit" value="INSERTAR COMENTARIO" tabindex="162">
</noscript>
```


Es decir, en caso de funcionar Javascript, se muestra un botón “Insertar comentario” que al ser pulsado ejecuta una función en Javascript que se encarga de evaluar todos y cada uno de los campos, comprobando si son correctos y si se pueden enviar a la siguiente página para que los procese. En caso de no funcionar, el botón que se muestra es simplemente un botón del tipo *submit* que envía los datos a la siguiente página donde mediante PHP se validarán.

Comprobación mediante Javascript, llamando a la función “validarComentario”:

```
function validarComentario(form){
    // nombre del recurso
    if (form.nombrerrecurso.value=="Seleccione")
        alert ("Debe elegir un valor para el campo \"Nombre del
        recurso\"")
    else
        // nombre de la sede
        if (form.nombresede.value=="Seleccione")
            alert ("Debe elegir un valor para el campo \"Nombre de la
            sede\"")
        else
            // comentario
            if ((form.comentario.value=="Escriba su comentario sobre este
            recurso")||(form.comentario.value==""))
                alert ("Debe introducir un valor en el campo
                \"Comentario\"")
            else
                form.submit()
}
```

Comprobación mediante PHP incrustado en el código HTML, una vez enviados los datos a la página que los recibe e interactúa con la base de datos:

```
<?php
    $nombreRecurso=$_POST["nombrerrecurso"];
    $nombreSede=$_POST["nombresede"];
    $comentario=$_POST["comentario"];

    if (($nombreSede=="Seleccione")||($nombreRecurso=="Seleccione"))
```

```
echo "<span class='aviso'>Debe seleccionar un recurso y una
sede.</span>";

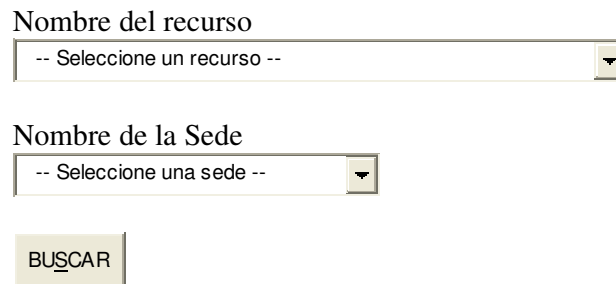
else if (($comentario=="")||($comentario=="Escriba su comentario
sobre este recurso"))
echo "<span class='aviso'>Debe escribir un comentario.</span>";

else{ // datos válidos
    // Procesar los datos recibidos
}
?>
```

El mismo proceso se pone en funcionamiento para cada uno de los formularios presentes en la aplicación Web.

➤ Actualización de menús desplegables

En muchas de las funcionalidades de la aplicación Web existen una serie de opciones que se deben elegir mediante menús desplegables. Por ejemplo, a la hora de modificar los datos de la sede de un recurso, se ha de elegir primero el recurso, luego alguna de la sedes que tiene ese recurso y, a continuación, pulsar el botón “Buscar” (Figura 7-3).



The image shows a web form with two dropdown menus and a button. The first dropdown menu is labeled "Nombre del recurso" and contains the text "-- Seleccione un recurso --". The second dropdown menu is labeled "Nombre de la Sede" and contains the text "-- Seleccione una sede --". Below these menus is a button labeled "BUSCAR".

Figura 7-3 Menús actualizados automáticamente

Para facilitar la tarea, se ha vinculado la selección de un recurso determinado, en el primer menú desplegable, con la actualización automática del segundo menú con todas aquellas sedes que tenga el recurso previamente seleccionado. Una vez escogido el recurso, el formulario se autoenvía a la misma página para captar mediante variables de formulario de PHP la selección hecha y, una vez conocido el recurso en cuestión, poder

conectar a la base de datos, buscar sus sedes, y situarlas en las opciones del siguiente menú desplegable. El proceso se expone a continuación.

En el primer menú desplegable se muestran todos aquellos recursos que existen en ese momento en la base de datos. Se captura el evento de selección de un recurso mediante el capturador de eventos “onChange”, de manera que se envía el formulario (compuesto únicamente por este campo).

```
<form method="post" action="modificar_sede.php">
<label for="recurso">
<span class="remarcado">Nombre del recurso</span>
</label>
<select name="nombrerecurso" id="recurso"
onChange="this.form.submit()">
<option value="Seleccione" selected>-- Seleccione un recurso --
</option>
<?php
    // Conexión con la BD
    ...
    // Recuperar todos los recursos
    ...

    while Hay_recursos()
    echo
        "<option value=\"nombre_recurso\">\".$nombre_recurso.</option>";
?>
</select>
</form>
```

Seguidamente se encuentra el otro menú desplegable, donde se debe hacer la selección de la sede del recurso anteriormente elegido que se quiere modificar. Este menú desplegable se encuentra enmarcado dentro de otro formulario que contiene como primer campo un elemento oculto. Mediante el uso de PHP y de la variable definida `$_POST` se obtiene el valor recibido a través del formulario y se le asigna al campo oculto. Esto permitirá que, pulsando “Buscar” y una vez enviado el formulario a la página siguiente que procesará los datos, se envíen tanto el nombre del recurso (presente en el campo oculto) como el nombre de la sede (presente en el menú desplegable).

```
<form method="post" action="modificar_sede.php?mostrar=si">

<input type="hidden" name="nombrerecurso" value="<?php echo
stripslashes($_POST["nombrerecurso"])?>">

<label for="sede">
<span class="remarcado">Nombre de la Sede</span>
</label>
<select name="nombresede" id="sede">
<option value="Seleccione" selected>-- Seleccione una sede --</option>
<?php
    // Conexión con la BD
    ...
    // Recuperar el recurso escogido
    $nombre_recurso=$_POST["nombrerecurso"];
    // Recuperar sus sedes
    ...
    while Hay_sedes()
        echo "<option value=\"nombre_sede\">".$nombre_sede."</option>";
?>
</select>

<input type="submit" value="BUSCAR">
```

Sin embargo, esta solución no parece válida para aquellos usuarios que dispongan de navegadores no compatibles con Javascript. Para tal caso, se ha optado por realizar el mismo proceso pero pulsando para ello un botón mediante el cual el recurso escogido se envíe igualmente y se puedan obtener todas sus sedes.

Este mecanismo puede comprenderse y verse fácilmente a través de la figura mostrada a continuación (Figura 7-4)

Nombre del recurso

-- Seleccione un recurso --

REFRESCAR SEDES

Nombre de la Sede

-- Seleccione una sede --

BUSCAR

Figura 7-4 Menús actualizados manualmente

El botón “Refrescar sedes” aparece únicamente cuando se detecta, mediante el elemento `NOSCRIPT` de HTML, que Javascript no es compatible con el navegador que se está utilizando. Al ser pulsado, se envía sin más permitiendo que el siguiente menú conozca cuál es el recurso escogido, pueda recuperar de la base de datos las sedes que le corresponden y mostrarlas como opciones. Se inserta para ello entre ambos menús el siguiente código HTML:

```
<noscript>
<input type="submit" value="REFRESCAR SEDES">
</noscript>
```

Este mismo esquema de funcionamiento se aplica en todas aquellas páginas de la aplicación Web que precisan del uso de varios menús desplegables, siendo las opciones de éstos cambiantes en función de los valores previamente escogidos.

6.4 “Para los scripts y applets, asegúrese de que los manejadores de evento sean independientes del dispositivo de entrada.” (Prioridad 2)

Los dispositivos de entrada no tienen por qué ser siempre un teclado y un ratón, aunque en la mayoría de los casos sea así. También pueden serlo dispositivos de braille, punteros de cabeza o micrófonos, entre otros muchos. Es por ellos que los manejadores de eventos utilizados deberían contemplar estas posibilidades de navegación.

Mediante los manejadores de eventos se puede invocar un *script* cuando determinada acción tiene lugar (se pulsa el ratón, se aprieta una tecla, se carga la página, se selecciona un opción de un menú, etc.). Si el manejador de evento se utiliza para lanzar un acción que realiza algo más importante que un cambio de apariencia o un efecto puramente decorativo, como puede ser un cálculo o un envío de un formulario, las pautas de accesibilidad recomiendan una serie de diseños:

- Utilizar manejadores de eventos a nivel de aplicación que no dependan del dispositivo de entrada, como “onFocus”, “onBlur” y “onSelect”.
- Si se deben utilizar atributos específicos de un dispositivo, proporcionar mecanismos redundantes de interacción. Por ejemplo, usan siempre de forma conjunta “onMouseDown” con “onKeyDown”, “onMouseup” con “onKeyUp” y usar “onClick” con “onKeyPress”.
- No emplear nunca eventos que dependan de las coordenadas del puntero del ratón.

En la aplicación Web de acceso a la base de datos de FEDACE se utiliza el manejador de evento “onKeyPress” siempre que aparece el manejador dependiente del ratón “onClick”. Así, por ejemplo, tenemos:

```
<input type="checkbox" name="tipo3" value="Centro"
onClick="comprobarCentro(this.form,8,31)"
onKeyPress="comprobarCentro(this.form,8,31)"> Centro
```

La acción aquí mostrada se ejecuta cuando el usuario selecciona o quita la opción “Centro” y habilita o deshabilita, respectivamente, aquellos campos del formulario que son exclusivos para los centros. Esta funcionalidad, empleada en otros formularios de la aplicación en caso de no tener activada la ejecución de Javascript, facilita el uso completo del formulario.

7.2.7 Pauta 7

“Asegure al usuario el control sobre los cambios de los contenidos temporales.”

Existen ciertos efectos de las páginas Web que convierten en una tarea casi imposible el que los dispositivos de navegación adaptados para las personas que presentan alguna discapacidad puedan funcionar correctamente en su cometido. Uno de los mayores impedimentos lo representa el movimiento dentro de la Web, ya sea de imágenes o de texto. Igualmente, aunque se esté usando un navegador convencional, personas que tienen algún problema visual o que les cuesta fijar la atención pueden encontrar incómodo el movimiento en el texto de la página. Lo mismo ocurre con las ventanas que se abren o cierran sin que el usuario pueda controlarlo o que se actualizan de forma automática cada cierto periodo de tiempo marcado por el creador de la Web, renovando sus contenidos o modificándolos.

En esta pauta no existe ningún punto de verificación aplicable a la aplicación Web en desarrollo puesto que no se barajó en ningún momento la posibilidad ni la necesidad de incluir páginas actualizadas automáticamente en su contenido, o ningún tipo de movimiento o parpadeo en los textos ofrecidos en ella.

7.2.8 Pauta 8

“Asegure la accesibilidad directa de las interfaces de usuario incrustadas.”

Muchas veces se incluyen en la aplicación Web objetos incrustados, como applets o *scripts*, que tienen su propia interfaz de uso, de la misma forma que el navegador o cualquier otro software poseen interfaz propia. En estos casos, siempre se debe diseñar la interfaz de forma que sea accesible igual que el resto de los elementos de la Web. En caso de no poder proporcionar esta accesibilidad, debe siempre facilitarse una solución alternativa accesible.

Esta pauta contiene únicamente un punto de verificación, que es el que se explica a continuación.

8.1 “*Haga los elementos de programación, tales como scripts y applets, directamente accesibles o compatibles con las ayudas técnicas*” (Prioridad 1 si la funcionalidad es importante y no se presenta en otro lugar; de otra manera, prioridad 2)

Como se ha explicado en pautas anteriores, la aplicación Web de gestión de la base de datos contiene una serie de *scripts* en Javascript que proporcionan funcionalidades especialmente de validación de datos de los formularios. Mediante el uso de PHP y manejadores de eventos no dependientes de dispositivos y redundantes, siguiendo además los criterios de diseño explicados anteriormente, se consigue que estos *scripts* se presenten accesibles.

7.2.9 Pauta 9

“Diseño para la independencia del dispositivo.”

No se debe caer en el frecuente error a la hora de diseñar una Web de que la mayoría de los usuarios que accedan a ella lo harán mediante navegadores tradicionales, un teclado y un ratón. Existen muchos usuarios con alguna discapacidad que utilizan en su navegación por la Web diferentes dispositivos, como sintetizadores de voz, punteros de cabeza (también llamados licornio) u otros muchos. Si algún elemento de la Web, como los campos de un formulario, sólo se puede activar mediante el ratón, cualquier persona que no use un dispositivo convencional será incapaz de poder utilizarlo.

Aquellos puntos de verificación que se pueden aplicar a la aplicación Web son los siguientes:

9.3 “*Para los scripts, especifique manejadores de evento lógicos en vez de manejadores de evento dependientes de dispositivos.*” (Prioridad 2)

En anteriores pautas y puntos de verificación, se ha mostrado la forma en la que se vincula la ejecución de un *script* a un manejador de eventos de forma redundante. Los manejadores de eventos permiten lanzar *scripts* cuando una determinada acción se produce. Los manejadores de evento lógicos son “onBlur”, “onFocus”, “onSelect” y los manejadores redundantes son “onKeyPress”, “onKeyUp” y “onKeyDown”.

9.4 *“Cree un orden lógico para navegar con el tabulador a través de vínculos, controles de formulario y objetos.”* (Prioridad 3)

Aquellos usuarios que presenten discapacidades motoras es posible que no puedan manejar dispositivos de apuntamiento como el ratón. En estos casos, el teclado puede convertirse en una herramienta indispensable para su navegación por la Web. Se deben proporcionar elementos que permitan, únicamente a través del teclado, el uso de las diferentes partes de la Web.

Mediante el elemento TABINDEX existente en HTML y la colocación lógica de los elementos de la aplicación Web de acceso a la base de datos, se permite el acceso a cada uno de sus componentes únicamente usando el tabulador del teclado. El orden establecido siempre es el siguiente:

1. Enlaces de la barra de navegación principal.
2. Enlaces de la barra de navegación secundaria.
3. Campos de los formularios.
4. Botones de acción de la página (para enviar, borrar, buscar, etc.).

9.5 *“Proporcione atajos de teclado para los vínculos más importantes (incluidos los de los mapas de imagen de cliente), los controles de formulario y los grupos de controles de formulario.”* (Prioridad 3)

Para facilitar la navegación por teclado a aquellos usuarios que necesiten hacer uso de ello para acceder a los distintos contenidos, se pueden establecer una serie de atajos para dirigir el foco de acción directamente a los elementos más importantes dentro de la Web.

HTML permite marcar este tipo de elementos mediante ACCESSKEY. De esta forma, pulsando simultáneamente la tecla ALT y la tecla asignada al elemento de la Web directamente seleccionable, se puede situar el foco en él. Dentro de la aplicación Web de gestión de la base de datos se ha utilizado este etiquetado especial para los elementos de la barra principal de navegación, siendo de la siguiente forma:

- Para el usuario Administrador:
 - Alt + C para “Consultar”
 - Alt + Z para “Actualizar”
 - Alt + U para la opción de “Usuarios”
 - Alt + E para la sección “Acceso”
 - Alt + O para “Contraseña”
 - Alt + S para “Salir”.

- Para el resto de usuarios, según sus permisos:
 - Alt + C para “Consultar recurso”
 - Alt + I para “Introducir comentario”
 - Alt + M para “Cambiar contraseña”
 - Alt + S para “Salir”
 - Alt + E para “Contáctenos”

Estas letras de acceso rápido aparecen siempre subrayadas para que de forma intuitiva y rápida el usuario las identifique como tales.

7.2.10 Pauta 10

“Utilice soluciones provisionales.”

De igual que forma que hay que pensar en aquellos usuarios que no utilizan los navegadores convencionales, a la hora de crear una Web también se debe tener en cuenta a los usuarios que no disponen de las últimas versiones lanzadas de los distintos navegadores. Esto es así porque pueden existir ciertas funcionalidades o comportamientos en los navegadores más modernos que no están disponibles en versiones anteriores de los mismos.

En esta pauta es importante resaltar que, tal y como explica el W3C en las “Pautas de accesibilidad al contenido en la Web 1.0” [27], *“Los puntos de verificación que contienen la frase ‘hasta que las aplicaciones de usuario...’ requieren que los desarrolladores de contenidos proporcionen soporte adicional para la accesibilidad*

hasta que la mayoría de las aplicaciones de usuario tengan disponibles para sus usuarios las necesarias características de accesibilidad”.

Los puntos de verificación aplicables en esta pauta son:

10.1 *“Hasta que las aplicaciones de usuario permitan desconectar la apertura de nuevas ventanas, no provoque apariciones repentinas de nuevas ventanas y no cambie la ventana actual sin informar al usuario.”*(Prioridad 2)

La apertura repentina de una nueva ventana sin que el usuario sea informado o tenga capacidad para bloquearlo dificulta la accesibilidad y provoca molestias innecesarias. Provoca pérdida de atención en el documento que previamente se estaba leyendo y además puede desorientar a aquellos usuarios que utilizan dispositivos especiales de navegación como, por ejemplo, lectores de pantalla para ciegos.

En el desarrollo de la aplicación Web de acceso a la base de datos se barajó la idea por parte del cliente de mostrar los comentarios hechos por otros usuarios en cada recurso mediante un enlace al final de los datos generales de éste que, al ser pulsado, desplegará una nueva ventana de menor tamaño con todos sus comentarios. Esta posibilidad, una vez estudiadas y analizadas estas pautas de accesibilidad, fue totalmente rechazada y justificada ante el cliente. En su lugar, los comentarios aparecen, debidamente diferenciados con un título y tipo de letra diferente, al final de los datos generales de los recursos.

10.2 *“Hasta que las aplicaciones de usuario soporten explícitamente la asociación entre control de formulario y etiqueta, para todos los controles de formularios con etiquetas asociadas implícitamente, asegúrese de que la etiqueta está colocada adecuadamente.”* (Prioridad 2)

Asociando a cada elemento del formulario una etiqueta se facilita su comprensión y la navegación entre todos ellos. El elemento LABEL de HTML, junto con los atributos FOR e ID, permiten realizar este etiquetado de formularios.

En todos los formularios presentes en la aplicación Web de gestión de la base de datos los elementos se han etiquetado de la forma correcta, siguiendo un patrón similar al mostrado a continuación:

```
<label for="presidencia">Presidencia</label>  
<input type="text" name="presidencia" id="presidencia"  
value="Presidencia" size="30" maxlength="30" >
```

10.4 *“Hasta que las aplicaciones de usuario manejen correctamente los controles vacíos, incluya caracteres por defecto en los cuadros de edición y áreas de texto.”*
(Prioridad 3)

Algunas versiones antiguas de ciertos navegadores impiden al usuario navegar por cuadros de texto editables que estén vacíos. Estos cuadros de texto son los elementos INPUT y TEXTAREA de los formularios.

Todos los elementos de este tipo dentro de la aplicación Web de gestión de la base de datos contienen algún valor por defecto que o bien explica qué valores se deben introducir o bien especifica el nombre del campo al que pertenecen. Además, para hacer más cómoda la navegación a aquellos usuarios que sí disponen de navegadores modernos, se ha implementado un *script* en Javascript que se ejecuta al situar o retirar el foco sobre el campo del formulario.

El funcionamiento es muy sencillo. Si al situar el foco de atención sobre el campo, éste contiene el texto por defecto en él introducido (por ejemplo, “Describa aquí el recurso” en el campo “Descripción”) entonces el *script* borrará de forma automática el texto por defecto para permitir al usuario introducir directamente el valor que desee. Si al retirar el foco del campo, éste contiene una cadena vacía (es decir, el usuario no ha introducido ningún valor), instantáneamente se escribirá de nuevo el texto por defecto correspondiente a este campo del formulario. No se realiza ninguna acción tanto si al situar el foco el campo tiene un valor anteriormente introducido por el usuario como si, al retirarlo, el campo contiene un valor que acaba de ser escrito también por el usuario.

Por ejemplo, en el caso de la descripción general de un recurso. He aquí el código HTML para este campo de formulario:

Descripción

```
<textarea name="descripcionrecurso"
onFocus="BorrarCampo(this.form,39,'Escriba aquí la descripción general
del recurso')" onBlur="LlenarCampo(this.form,39,'Escriba aquí la
descripción general del recurso')" rows="15" cols="60">
Escriba aquí la descripción general del recurso
</textarea>
```

Mediante los manejadores de eventos lógicos “onFocus” y “onBlur” se detecta, respectivamente, si el usuario sitúa el foco de acción en el campo o lo retira. En cada caso se llama a una función en Javascript que recibe tanto el elemento del formulario (en este caso el campo “Descripción”) como el texto que por defecto lleva asociado este campo. Las funciones son las siguientes:

```
function BorrarCampo(form,num,cadena){
    if (form.elements[num].value==cadena)
        form.elements[num].value=""
}

function LlenarCampo(form,num,cadena){
    if (form.elements[num].value=="")
        form.elements[num].value=cadena
}
```

En cada caso, simplemente comprueban el valor actual del campo recibido para realizar la acción correspondiente (borrarlo o introducir la cadena por defecto).

10.5 *“Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten claramente los vínculos contiguos, incluya caracteres imprimibles (rodeados de espacios), que no sirvan como vínculo, entre los vínculos contiguos.”* (Prioridad 3)

Es muy común, sobre todo cuando se tratan de barras de navegación o enlaces a páginas de interés, situar de forma contigua enlaces que llevan a lugares distintos y que no tienen relación entre sí. Algunos lectores de pantalla no realizan ninguna pausa si no

encuentran algún carácter imprimible entre ellos, por lo que su inclusión ayudará a los usuarios invidentes a navegar por la Web. Igualmente, estos caracteres de separación pueden ayudar a distinguir los enlaces entre sí a los restantes usuarios.

La barra de navegación principal de la aplicación Web de acceso a la base de datos contiene una serie de enlaces que se sitúan de forma contigua. Para hacer una separación entre ellos, se ha optado por incluir el carácter “|” rodeado a ambos lados de un espacio en blanco. Así, queda de la siguiente forma (Figura 7-5):



Figura 7-5 Barra de navegación principal con caracteres imprimibles entre enlaces

7.2.11 Pauta 11

“Utilice las tecnologías y pautas W3C.”

El objetivo principal del W3C desde el momento de su creación es el de *“guiar la Web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la Web”* [26]. El W3C no está guiado por intereses comerciales o de carácter empresarial, su grupo de trabajo está constituido por diversos especialistas en diferentes áreas que, mediante amplio consenso, emiten documentos de diferentes tipos para estandarizar el desarrollo de la Web.

Toda recomendación lanzada desde el W3C debería ser aplicada conjuntamente con las pautas sobre accesibilidad dictadas desde este mismo organismo. En el caso extremo de no poder ofrecer una solución accesible, el último recurso al que se debe recurrir es el de crear una página alternativa con los mismos contenidos de la página principal. Se debe, en tal caso, prestar muchísimo interés en actualizarla con la misma frecuencia que la página no accesible, puesto que algo tanto frustrante (e incluso ofensivo) como una Web no accesible es una Web alternativa con contenidos no actualizados.

Los puntos de verificación aplicables son los siguientes:

11.1 *“Utilice tecnologías W3C cuando estén disponibles y sean apropiadas para la tarea y use las últimas versiones que sean soportadas.”* (Prioridad 2)

Las pautas de accesibilidad sugieren siempre utilizar tecnologías aprobadas y revisadas desde el W3C. Estas tecnologías, una vez aplicadas, deben ser validadas por los servicios de validación automática certificados por el consorcio.

Las tecnologías utilizadas en la construcción de la aplicación Web son básicamente dos: HTML, en su versión 4.01 *Transitional*, y hojas de estilo o CSS. En ambos casos, se ha utilizado el validador automático existente en el W3C. [42]

11.2 *“Evite características desaconsejadas por las tecnologías W3C.”* (Prioridad 2)

En el empleo de las tecnologías recomendadas desde el W3C debe ir además acompañado de una especial atención en no usar aquellos elementos que, quizás en versiones anteriores o preliminares, estaban también recomendados o disponibles, pero que posteriores revisiones han sido desaconsejados.

Por ejemplo, dentro de este grupo de elementos desaconsejados en HTML se encuentran todos los relacionados con la apariencia de los datos. Atributos como FONT, COLOR, BGCOLOR, LINK, VLINK, ALINK, y un largo etc. están totalmente desaconsejados y no deben ser utilizados. En lugar de ello se recomienda encarecidamente, con el importante fin de separar definitivamente la estructura de la apariencia en la Web, el uso de hojas de estilo o CSS. Es el caso de la aplicación Web de acceso a la base de datos, donde en ningún momento se usa ninguno de los elementos desaconsejados y, en el caso de los elementos relacionados con la apariencia, se han sustituido por propiedades en las hojas de estilo definidas.

11.4 *“Si, después de los mayores esfuerzos, no puede crear una página accesible, proporcione un vínculo a una página alternativa que use tecnologías W3C, sea accesible, tenga información (o funcionalidad) equivalente y sea actualizada tan a menudo como la página (original) inaccesible.”* (Prioridad 1)

La posibilidad de crear una versión alternativa con contenidos accesibles debe ser siempre la última solución a la que recurrir. Se deben dedicar todos los esfuerzos posibles en desarrollar una Web plenamente accesible y común para todos los usuarios.

La actual Web de FEDACE [5] presenta información acerca de los principios y objetivos básicos de la organización, sus trabajos con las asociaciones federadas y explica en qué consiste el daño cerebral y sus consecuencias. No es un sitio Web excesivamente complejo, ya que no cuenta con elementos multimedia ni objetos incrustados. Sin embargo, no se siguieron las pautas mínimas de accesibilidad durante su diseño, construyendo en su lugar una versión de sólo texto plano que pretende ser una alternativa accesible a la Web original.

Desde el primer momento en el diseño e implementación de la aplicación Web de gestión a la base de datos sobre recursos de daño cerebral adquirido, se puso especial hincapié en materia de accesibilidad. Las “Pautas de accesibilidad al contenido en la Web” [27] han sido una guía constante desde el principio de este proyecto hasta el final, buscando siempre construir una única aplicación Web con todos y cada uno de sus componentes accesibles para todos los usuarios.

7.2.12 Pauta 12

“Proporcione información de contexto y orientación.”

A la hora de diseñar una aplicación Web accesible, es importante facilitar con todos los medios disponibles la navegación y comprensión del contenido de la misma. Toda información orientativa o de contexto que se facilite será bienvenida por todos los usuarios, especialmente por aquellos que presenten alguna discapacidad.

A continuación se muestran los puntos de verificación aplicables a la aplicación Web diseñada:

12.3 “Divida los bloques largos de información en grupos más manejables cuando sea natural y apropiado.” (Prioridad 2)

Por ejemplo, un caso claro de agrupación de información es cuando se vinculan de forma lógica un conjunto de elementos de formulario por formar parte de un único concepto o unidad de información. Con este fin, se pueden utilizar los elementos FIELDSET y LEGEND de HTML.

En la aplicación Web de gestión de la base de datos en numerosas ocasiones se agrupan elementos de un formulario mediante estas etiquetas. Así, por ejemplo, cuando se pide el tipo de recurso se tiene el siguiente código en HTML:

```
<fieldset>
<legend>Tipo de recurso</legend><br>
<input type="checkbox" name="tipo1" value="Asociación de
afectados">Asociación de afectados<br>
<input type="checkbox" name="tipo2" value="Asociación de
profesionales">Asociación de profesionales<br>
<input type="checkbox" name="tipo3" value="Centro">Centro<br>
</fieldset>
```

12.4 “Asocie explícitamente las etiquetas con sus controles.” (Prioridad 2)

Como ya se ha mencionado anteriormente, en los campos de los formularios se deben asociar de forma correcta etiquetas de control que ayuden a navegar por ellos y a comprender su significado. Mediante el elemento LABEL y los atributos ID y FOR se consigue cubrir este punto de verificación.

En la aplicación Web desarrollada en el presente proyecto se han asociado a cada campo de los muchos formularios existentes las etiquetas correspondientes, como por ejemplo:

```
<label for="web">Web</label>
<input type="text" value="http://" name="web" id="web" size="30"
maxlength="60">
```

7.2.13 Pauta 13

“Proporcione mecanismos claros de navegación.”

Es importante ayudar a la navegación por la Web mediante información orientativa, barra de navegación, mapas del sitio, etc., ayudando así a los usuarios a encontrar aquello que están buscando. Además, las personas con discapacidad visual o cognitiva dispondrán de mayores posibilidades de acceder con éxito a los contenidos ofrecidos.

Los puntos de verificación aplicables en esta pauta son:

13.1 *“Identifique claramente el objetivo de cada vínculo.”* (Prioridad 2)

Los enlaces de hipertexto constituyen la principal herramienta de navegación y son uno de los elementos más comunes dentro de una página Web. Es por ello muy importante facilitar toda la información posible acerca del contenido al que se accede cuando son pulsados. Un texto adecuado para un vínculo no debería ser demasiado general o carente de significado, como el popular pero poco recomendado “pinche aquí”, sino una brevísima descripción de cuál es su documento vinculado. Igualmente, mediante el etiquetado correcto se puede proporcionar información adicional, al usar por ejemplo el elemento TITLE de HTML. La navegación entre enlaces mediante TABINDEX será también agradecida por muchos usuarios.

En la aplicación Web de gestión de la base de datos de FEDACE se han utilizado siempre textos descriptivos para los vínculos. Los vínculos resumen, en pocas palabras, su cometido, como por ejemplo “Volver al principio de la página” o “Contáctenos”.

Además, todos los enlaces llevan un atributo TITLE que describe mejor cuál es su cometido:

```
<a href="consultar_recurso.php" title="Consultar recursos, usuarios y  
comentarios (Alt+C)" accesskey="C">Consultar</a>
```

13.2 “Proporcione metadatos para añadir información semántica a las páginas y sitios.” (Prioridad 2)

Los metadatos en HTML, o información sobre los datos, si están bien contruidos pueden proporcionar a los usuarios una ayuda orientativa muy útil sobre el documento que están visitando. En ellos se incluyen elementos como TITLE, para dar un título descriptivo al documento; META, que proporciona toda clase de información extra (como autor, palabra clave, descripción del documento extendida, etc.); o la versión de la tecnología usada en su construcción mediante DOCTYPE.

Todas las páginas que forman parte de la aplicación Web del presente proyecto contienen los siguientes metadatos:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<head>
<title>FEDACE Guía de Recursos sobre Daño Cerebral Adquirido </title>
<META name="title" content="FEDACE Guía de Recursos sobre Daño
Cerebral Adquirido">
<META name="description" content="Guía de asociaciones de afectados y
familiares y centros médicos para afectados de daño cerebral
adquirido, ofrecida por la Federación Española de Daño Cerebral
FEDACE.">
<META name="keywords" content="fedace,daño cerebral,dca,daño cerebral
adquirido,guía recursos,asociación afectados,centro médico,">
<meta HTTP-EQUIV="content-type" CONTENT="text/html; charset=ISO-8859-
1">
<META name="language" content="es">
<META name="revisit-after" content="6 month">
<META name="rating" content="General">
<META name="author" content="Jaime Aranaz">
<META name="owner" content="FEDACE">
<META name="robot" content="index, follow">
</head>
```

13.4 “*Utilice los mecanismos de navegación de forma coherente.*” (Prioridad 2)

Los mecanismo de navegación, como una barra con los enlaces principales, facilitan el movimiento del usuario dentro de la página y ayudan a que éste pueda encontrar fácilmente lo que está buscando. Una barra de navegación, por ejemplo, crea un camino que los usuarios pueden recorrer en su utilización de la Web. Es necesario que estas barras de navegación sean lo suficientemente claras y explícitas para no perder a los usuarios menos adaptados a la navegación visual por la Web o a los que utilizan dispositivos de navegación especiales.

La aplicación Web de gestión de la base de datos permite la navegación de los usuarios mediante dos barras de navegación:

- Una barra de navegación principal, situada en la parte superior de forma horizontal, donde están siempre presente las principales secciones de la aplicación Web.
- Una barra de navegación secundaria, situada en la parte izquierda de forma vertical, donde se informa sobre las diferentes opciones existentes dentro de la sección principal actualmente seleccionada.

13.6 “*Agrupe los vínculos relacionados, identifique el grupo (para las aplicaciones de usuario) y, hasta que las aplicaciones de usuario lo hagan, proporcione una manera de evitar el grupo.*” (Prioridad 3)

Cuando un conjunto de vínculos contiguos pertenecen a una misma funcionalidad o cometido y pueden ser agrupados de forma lógica, se ayuda al usuario a comprender qué documentos encontrará al pulsarlos y a navegar fácilmente por ellos. Con el elemento MAP y el atributo TITLE se pueden agrupar aquellos enlaces que tengan alguna relación entre ellos.

Por ejemplo, dentro de la aplicación Web de acceso a la base de datos, se agrupan aquellos enlaces que pertenecen a la barra de navegación secundaria de la siguiente forma mostrada. El ejemplo pertenece a los enlaces de la opción “Actualizar”:

```
<map title="Insertar elementos" name="Insertar elementos">
<span class="remarcado">INSERTAR</span><br>
<a href="insertar_recurso.php" title="Insertar un nuevo recurso"
class="menu">Recurso</a><br>
<a href="insertar_sede.php" title="Insertar una sede"
class="menu">Sede</a><br>
<a href="insertar_contacto.php" title="Insertar una persona de
contacto" class="menu">Contacto</a><br>
<a href="insertar_comentario.php" title="Insertar un comentario"
class="menu">Comentario</a><br>
</map>
```

7.2.14 Pauta 14

“Asegúrese de que los documentos sean claros y sencillos.”

Mediante esta pauta se busca simplemente que el lenguaje utilizado y la estructura básica de la página sea lo más simple y clara posible, con el fin de facilitar la comprensión del contenido y la búsqueda de aquella información que se necesite. Técnicas como utilizar vínculos con sentido y autoexplicativos, no utilizar lenguaje demasiado técnico (a no ser que el tema lo requiera) ni redundante, ser conciso y claro en las explicaciones o el uso de encabezados ayudarán a todo tipo de usuarios a encontrar la Web sencilla y agradable de utilizar.

14.3 “Cree un estilo de presentación que sea coherente para todas las páginas.” (Prioridad 3)

Usar un estilo de presentación coherente en toda la Web, permite a los usuarios localizar los mecanismos de navegación más fácilmente. Las personas con discapacidad mejorarán el aprendizaje y la lectura, pero éstas prácticas también facilitan la navegación a todos los usuarios. Si la navegación es más predecible, aumentará la probabilidad de que el usuario encuentre la información en un sitio o la evite si así lo desea.

La estructura básica del contenido en toda la aplicación Web es la siguiente:

1. Barra de navegación principal

2. Barra de navegación secundaria
3. Contenido de la opción actual
4. Botones de acción (si procede)

Así mismo, mediante el uso de hojas de estilo se obtiene la apariencia del contenido, no mezclando en ningún momento estructura y formato. Las hojas de estilo se aplican a lo largo de toda la aplicación Web dando un aspecto uniforme y compacto en todas sus páginas, ayudando al usuario a identificar cada elemento como algo perteneciente a la mismo sitio Web y facilitando la identificación de los contenidos.

7.3 Validación de la accesibilidad

A pesar de haber aplicado las pautas anteriormente descritas durante las fases de diseño e implementación de la aplicación Web, es necesario comprobar que verdaderamente se está obteniendo un recurso Web accesible. Para ello, la mejor herramienta (pero no la única) son los validadores automáticos que existen disponibles en la red para todos los desarrolladores, acompañándolos siempre de una revisión manual.

7.3.1 Validación mediante TAW

La herramienta TAW es un útil validador automático de la accesibilidad desarrollado por la fundación CTIC [43] y financiado por el Ministerio de Industria, Turismo y Comercio, así como por el Gobierno del Principado de Asturias. Se basa en las reglas desarrolladas por el WAI [3] y en las pautas sobre accesibilidad dictadas desde el W3C.

El TAW funciona de forma remota, introduciendo una URL para ser analizada, o de forma local si se descarga una versión instalable del software. Para el desarrollo del presente proyecto, se optó por descargar este software e instalarlo en el equipo informático donde se estaba llevando a cabo la implementación de la aplicación Web de acceso a la base de datos. El uso dado a este validador automático ha sido el de ir comprobando *in situ* la accesibilidad de cada página de la aplicación Web al mismo tiempo que estaba siendo implementada. De esta forma, se iban corrigiendo al mismo

tiempo aquellos detalles pasados por alto y que impedían que la página correspondiente alcanzase el nivel de conformidad AAA, el objetivo principal.

Las páginas contenían en muchas ocasiones código en PHP que impedían el que pudieran ser directamente comprobadas por TAW. El proceso que se debía seguir era el de colgar la página en cuestión en el servidor Web instalado en el equipo de implementación y visionarla desde allí a través de cualquier navegador. Una vez procesada por el servidor, el código PHP se ejecuta y es enteramente traducido a HTML, de forma que al obtener el código fuente de la página no se obtiene nada más que código HTML, procesable por TAW.

En las primeras páginas implementadas fue donde se encontraron más errores de accesibilidad, sobre todo con las puntos de verificación relacionados con el etiquetado de los componentes de los formularios y con la accesibilidad de los *scripts*. Una vez completadas las primeras páginas, los errores fueron disminuyendo gracias a la mayor soltura en su desarrollo y previsión de los mismos basándose en los casos anteriores.

Una vez implementada la aplicación Web por completo, se consiguió que el validador TAW le otorgara una calificación de nivel AAA.

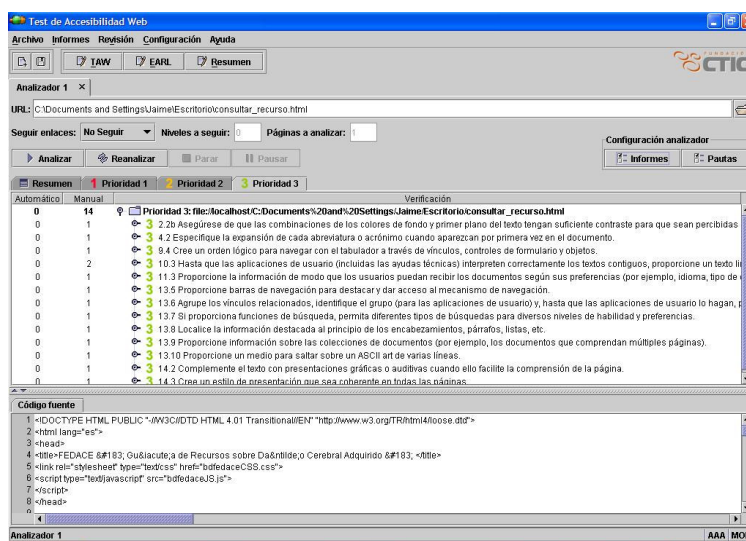


Figura 7-6 Interfaz de TAW

La interfaz presentada por TAW (Figura 7-6) en su versión instalable es muy simple y concisa, permitiendo realizar de forma rápida el análisis de cualquier documento de forma local. Tras el análisis, se ofrece un resumen del número de puntos de verificación que han resultado fallidos y de los puntos de verificación manuales, ordenados según prioridad. Este resumen se puede estudiar también de forma más detallada para saber qué punto de verificación exactamente se ha fallado. También se visiona el código fuente de manera que sea fácil localizar la línea exacta donde algún punto de verificación ha dado negativo. TAW permite la elaboración de informes de la validación hecha, así como seguir los vínculos presentes en la Web para extender el análisis de accesibilidad a los documentos asociados.

7.3.2 Validación mediante herramienta HERA

El otro validador automático utilizado es HERA [29], cuya versión actualmente disponible es la 2.0 beta. Esta herramienta debe usarse introduciendo directamente una URL para ser analizada. Elabora análisis muy completos, indicando de cada prioridad los puntos de verificación aplicados y pasados con éxito, los que han fallado y además aquellos puntos que no pueden ser aplicados a la Web en cuestión. También informa acerca de las importantes revisiones manuales a realizar para completar la revisión de la accesibilidad.

Como ayudas para la revisión, HERA expone de forma detallada cada punto de verificación de todas las pautas marcadas desde el W3C, explicando en qué consisten y señalando en el propio código de la página las líneas exactas donde se encuentran los elementos correspondientes a ese punto de verificación. Permite, así mismo, generar informes acerca de la revisión llevada a cabo en formatos PDF, RDF y XHTML. Un ejemplo de la interfaz usada por HERA puede comprobarse en la Figura 7-7.

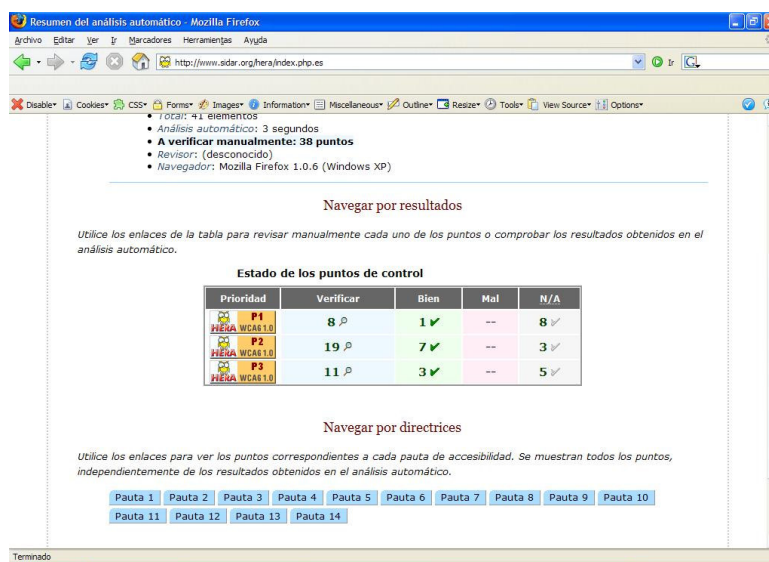


Figura 7-7 Interfaz Web de HERA

HERA se utilizó para validar la accesibilidad de la aplicación Web de gestión de la base de datos una vez que la implementación de sus páginas se había completa y, de forma local, se habían revisado mediante TAW. La aplicación Web fue colgada en el servidor instalado en un equipo de prueba con conexión a Internet y vinculada a un dominio adquirido de forma gratuita por la empresa DynDNS [37]. De esta forma, se fue comprobando una a una cada una de las páginas que componían la aplicación Web. El resultado fue que ciertos puntos de verificación que sí validaba TAW, HERA advertía que se estaban cometiendo errores en ellos. Tal fue el caso, por ejemplo, con los enlaces contiguos que no contenían caracteres imprimibles entre ellos (fallo no detectado por TAW).

Se podría decir, por lo tanto, que HERA representa un validador automático más completo y exigente que TAW, y que, de la misma manera que se recomienda repasar los puntos de verificación manuales para asegurar una completa accesibilidad, también es recomendable utilizar más de una única herramienta de validación automática para certificar que se está ante una Web plenamente accesible.

7.3.3 Validación de HTML

Todas las páginas de la aplicación Web de gestión de la base de datos están construidas bajo el lenguaje HTML, haciéndose por lo tanto necesario validar el etiquetado realizado a través de las herramientas correspondientes.

Desde el W3C se ofrece un servicio de validación automático [42] o bien mediante la introducción de la URL donde está alojada la Web o bien mediante la subida de ficheros donde se encuentra el código fuente de la página. La validación fue exitosa y certifico la correcta utilización de HTML 4.01 *Transitional* en todas las páginas de la aplicación. Tal situación es anunciada en la aplicación Web mediante el logotipo correspondientes.

7.3.4 Validación de CSS

Al igual que HTML, las hojas de estilo o CSS utilizadas requieren ser validadas para cubrir otro importante punto de verificación dentro de la pautas de accesibilidad, además de seguir la recomendación de utilizar siempre gramáticas formales.

La herramienta de validación automática del W3C utilizada [42] confirmó que las hojas de estilo utilizadas en la aplicación Web de acceso a la base de datos se corresponden con los estándares dictados desde este mismo organismo. Esta circunstancia se constata en la aplicación a través de la colocación de un icono identificativo.

7.3.5 Uso de diferentes navegadores

Una de las recomendaciones del W3C a la hora de validar la accesibilidad es el uso de diferentes navegadores y versiones para realizar pruebas. Esto es así puesto que, aunque existen uno o dos navegadores que son los que reúnen la mayor cuota de mercado entre los usuarios de Internet, existen múltiples navegadores disponibles y los usuarios pueden cambiar con frecuencia de navegador favorito, según éstos se van actualizando con nuevas funcionalidades más atractivas.

Es por ello que para probar la aplicación Web de gestión de la base de datos se utilizaron una serie de navegadores actualmente extendidos para comprobar que el funcionamiento era igual en todos y cada uno de ellos:

- Mozilla Firefox 1.0.6
- Internet Explorer 6.0
- Opera 8.5

El resultado fue en su mayoría satisfactorio, puesto que todos los navegadores parecían funcionar de la misma forma con la aplicación Web, aunque las únicas incidencias se produjeron bajo el navegador Internet Explorer. Ciertas implementaciones en Javascript no funcionaban bajo este navegador (acceso a los caracteres de una cadena mediante índice y no la función definida *charAt()*) y tuvieron que ser realizadas de otra forma. Así mismo, las hojas de estilo tuvieron que adaptarse a Internet Explorer ya que el estilo definido para las etiquetas de los controles de formulario aparecían con una sangría excesiva.

8 PLANIFICACIÓN FINAL Y ANÁLISIS DE COSTES

Mientras que la planificación inicial se realiza en las etapas inmediatas al comienzo del proyecto y pretende servir de guía durante la realización del mismo, en la planificación final se plasman los tiempos de desarrollo reales, costes y distribución en el tiempo del proyecto una vez que éste ha finalizado. Así mismo, permite hacer una comparación con la planificación inicial, qué atrasos o desajustes se han producido y prevenirlos para planificaciones futuras en proyectos que tenga un desarrollo similar.

8.1 Planificación final

La estimación inicial realizada sobre el tiempo de realización del presente proyecto se vio pronto afectada y de forma significativa debido la concesión de una beca de colaboración en el proyecto de investigación CESyA. Desde enero de 2006 hasta junio de ese mismo año se estuvo dedicando a dicha beca una media de 5 horas diarias y por ello se debió alterar el calendario en un principio propuesto. El tiempo final de desarrollo del proyecto pasó de los 5 meses inicialmente estimados a los 7 meses reales.

El deseo de las dos partes involucradas, cliente y equipo de desarrollo, era el de establecer como fecha límite para la finalización del proyecto el mes de junio. Por lo tanto, las medidas que se tomaron para poder cumplir con los plazos fueron las siguientes:

- La jornada laboral se redujo de 8 a 5 horas de lunes a viernes debido a la beca desarrollada en CESyA.
- Los sábados se estableció una jornada laboral de 8 horas, aplicándoles el salario correspondiente a horas extraordinarias.
- Se suprimieron los días festivos siguientes: en el mes de Marzo, los días 13 y 14; en el mes de Mayo, los días 1, 2 y 15. La jornada en dichos días fue de 5 horas y se les aplicó el salario de horas extraordinarias.
- Los domingos se mantuvieron como festivos.

Las siguientes figuras muestran la planificación final (Figura 8-1) y el diagrama GANTT correspondiente (Figura 8-2).

Figura 8-1 Planificación final

Figura 8-2 Diagrama GANTT final

8.2 Análisis de costes

Los costes derivados de la realización del proyecto han tenido que ser actualizados no sólo en el número de horas invertidas con respecto a la planificación inicial, sino también en la aplicación de horas extraordinarias que se han trabajado para cumplir con los plazos de entrega acordados.

El coste del proyecto por lo tanto aumenta considerablemente, no ya sólo por el mayor número de horas empleadas en la consecución del proyecto de las inicialmente previstas en la planificación inicial, sino también porque una parte considerable de estas horas han de pagarse sustancialmente más por tratarse de horas extraordinarias.

Así, basándose en los criterios utilizados a día de hoy [46], los salarios aplicados a cada miembro del equipo son los que siguen (Figura 8-3):

Cargo	Salario ordinario	Salario extraordinario
Jefe de Proyecto	25 €/hora	40 €/hora
Analista de BD	20 €/hora	30 €/hora
Analista Web	20 €/hora	30 €/hora
Diseñador de BD	20 €/hora	30 €/hora
Diseñador Web	20 €/hora	30 €/hora
Programador de BD	15 €/hora	20 €/hora
Programador Web	15 €/hora	20 €/hora

Figura 8-3 Salario ordinario y extraordinario del equipo

El tiempo total en horas ha sido de 937,7 y el coste asociado final de 18708,5 euros. La distribución final del equipo y las horas empleadas en las distintas fases se describe de la siguiente manera (Figura 8-4).

Etapa	Duración (horas)	Recursos			Coste (€)
		Empleado	Horas ord.	Horas extra	
Toma de requisitos	79,8	Jefe del Proyecto	27,5	6,2	1908,5
		Analista BD	14,25	3,8	
		Analista Web	14,25	3,8	
		Diseñador Web	5	0	
		Programador Web	5	0	
Análisis de requisitos	195,5	Analista BD	57,5	23	4440
		Analista Web	85	30	
Diseño global	76,4	Diseñador BD	25	15,7	1842
		Diseñador Web	20	15,7	
Implementación versión 1.0	210,5	Diseñador Web	15	6	3535,5
		Programador BD	25	4	
		Programador Web	122,5	38	
Pruebas versión 1.0	66	Diseñador BD	10	0	1300
		Diseñador Web	20	8	
		Programador Web	15	8	
		Programador BD	5	0	
Instalación versión 1.0	4,75	Jefe de Proyecto	1,25	0	83,75
		Programador BD	1,75	0	
		Programador Web	1,75	0	
Comprobar nivel de satisfacción	71	Jefe de Proyecto	17,5	4	1387,5
		Analista BD	8,75	2	
		Analista Web	8,75	2	
		Diseñador BD	10	4	
		Diseñador Web	10	4	
Implementación versión 2.0	150	Programador BD	15	8	2450
		Programador Web	95	32	
Pruebas versión 2.0	56	Diseñador BD	7,5	4	1152,5
		Diseñador Web	15	8	
		Programador Web	12,5	4	
		Programador BD	5	0	
Instalación versión 2.0	4,75	Jefe de Proyecto	1,25	0	83,75
		Programador BD	1,75	0	
		Programador Web	1,75	0	
Comprobar nivel de satisfacción	15	Jefe de Proyecto	5	0	325
		Analista BD	2,5	0	
		Analista Web	2,5	0	
		Diseñador BD	2,5	0	
		Diseñador Web	2,5	0	
Entregar versión final	8	Jefe de Proyecto	8	0	200
TOTAL HORAS	937,7	TOTAL COSTE			18708,5

Figura 8-4 Distribución final del equipo y su coste en las diferentes etapas

Como se puede constatar, el número de horas empleadas en cada una de las etapas también ha cambiado, en algunos casos sustancialmente. Así, se puede comprobar cómo en todas las etapas de desarrollo, en mayor o menor medida, se emplearon más horas de las inicialmente previstas. El incremento en horas se produjo especialmente en las fases de implementación de cada una de las versiones entregadas.

Un dato significativo es la diferencia de horas en la etapa de diseño entre la planificación inicial y la final. Éste es el único caso donde se emplearon menos horas de las en un principio previstas. La explicación más obvia para esta anomalía es que el trabajo desempeñado en CESyA abarcaba muchas tareas en casos similares de análisis y diseño que las luego desempeñadas en el presente proyecto. Una mayor soltura, adquirida por esta misma actividad en CESyA, ha propiciado una reducción de horas en este sentido.

Es igualmente fácil imaginar que otras muchas tareas hubieran llevado finalmente más tiempo de no haber trabajado en CESyA colaborando en un proyecto muy similar al presente. Es decir, si por un lado la beca en CESyA varió la fecha de entrega y finalización del proyecto, obligando a introducir horas extras de trabajo y varias semanas más de desarrollo, por el otro es cierto que el trabajo desempeñado en CESyA ha proporcionado un mayor conocimiento y habilidad en los campos abarcados por este proyecto.

9 CONCLUSIONES Y TRABAJOS FUTUROS

En la etapa final de este documento, una vez expuestos todos los trabajos llevados a cabo en el desarrollo completo de la base de datos y la aplicación Web accesible para su gestión, se detallan a continuación las conclusiones finales del proyecto y los posibles trabajos futuros.

9.1 Conclusiones finales

Se puede decir que los objetivos marcados al principio se han cumplido satisfactoriamente. La base de datos de recursos sobre daño cerebral supondrá un ahorro de trabajo para FEDACE y además, mediante la aplicación Web accesible de gestión y consulta podrá ofrecer información sobre los recursos a todas las asociaciones y afectados vinculados a esta federación.

Además, el proyecto ha permitido no sólo afianzar los conocimientos aprendidos durante la carrera, sino también adquirir otros muchos. Entre todos ellos, cabe destacar las pautas de accesibilidad para lograr acercar a las personas con alguna discapacidad el emergente e importante mundo que comienza a ser ya la Web. No sólo se conocen las técnicas a aplicar gracias a este proyecto, sino que además se ha tomado conciencia de lo difícil que a veces nos empeñamos en ponerles las cosas a aquellos que son diferentes. Se puede afirmar con seguridad que, a partir de ahora, el diseño e implementación de cualquier aplicación Web que sea llevada a cabo por el autor del presente proyecto tendrá una sensibilidad especial para lograr que la accesibilidad sea un objetivo primordial y básico de la misma.

También se ha podido confirmar que realizar un buen proyecto software no significa simplemente aplicar todos los conocimientos aprendidos en clase de la forma enseñada o disponer de los medios técnicos necesarios. Significa igualmente saber lidiar con los problemas que de forma inevitable surgen, encontrar la solución más adecuada y ponerla en práctica, no desmotivarse cuando parece que todo está saliendo mal y no como uno lo había planeado y que, como en casi todas las facetas de la vida, el entorno humano vinculado al proyecto (ya sea el cliente, el tutor o el entorno familiar o social

del propio autor) es una variable muy importante y que puede marcar de forma irreversible el éxito o el fracaso del mismo. Esto último se ha podido comprobar de forma directa y muy positiva tanto durante el desarrollo del presente proyecto como durante el tiempo en el que se estuvo trabajando en el proyecto CESyA.

9.2 Trabajos futuros

La experiencia de numerosos desarrolladores de sitios Web enfocados a una plena accesibilidad les permite afirmar que conseguir una aplicación Web plenamente accesible con calificación AAA es una tarea más complicada de lo que en un principio pueda parecer. Los validadores automáticos de accesibilidad no pueden ser la única herramienta a la hora de certificar la accesibilidad. Es preciso, además, revisar de forma manual, uno por uno y concienzudamente, cada punto de verificación para comprobar, según la situación y el contexto, si realmente se cumplen. La experiencia y el dominio sobre la materia del diseñador son fundamentales para la revisión completa de una Web accesible.

Una comprobación importante que debería realizarse en toda aplicación Web que pretenda ser accesible, como es el caso de la aplicación Web de gestión de la base de datos, es la prueba real con personas con alguna discapacidad. De esta forma, se conocen de primera mano los fallos o elementos que dificultan verdaderamente la accesibilidad. Además, durante la finalización de este proyecto, se ha conocido la existencia de otro validador automático, llamado “Examinator” [44], que va cobrando relevancia debido a su alto nivel de exigencia.

Por otro lado, el actual sitio Web del que dispone FEDACE con información acerca de sus actividades y objetivos como federación, como se ha venido mencionando a lo largo del presente documento, tiene una versión alternativa y accesible a la Web principal. Esta opción es válida pero es únicamente recomendada como la última opción a la que recurrir en caso de que no sea posible conseguir unos contenidos accesibles en la Web principal. Un diseño más concienzudo hubiera permitido realizar una única Web accesible sin demasiados problemas. Además, FEDACE tiene importantes planes de futuro, como es la puesta en marcha de un completo portal Web sobre el daño cerebral,

pionero en España y en Europa, donde poder prestar un servicio mucho más cercano y personalizado a los afectados por dicha dolencia. El proyecto está en fases preliminares y a la espera de que le sea concedida una subvención por parte del Ministerio de Trabajo y Asuntos Sociales. Como trabajo futuro se encuentra el de realizar un análisis y un diseño de este portal aplicando toda la experiencia y los conocimientos adquiridos en materia de accesibilidad.

También se podría barajar la idea de actualizar la base de datos con información pormenorizada de cada una de las personas que forman parte del equipo médico en las sedes de los recursos. Esto permitiría crear un fondo sobre profesionales y su experiencia muy útil para los afectados por el daño cerebral.

10 GLOSARIO DE TÉRMINOS

- **Accesibilidad:** La accesibilidad indica la facilidad con la que algo puede ser usado, visitado o accedido en general por todas las personas y especialmente por aquellas que poseen algún tipo de discapacidad.
- **ANSI/SPARC:** es un grupo de estudio del *Standard Planning and Requirements Committee* (SPARC) del ANSI (siglas del Instituto Nacional Americano de Estándares, *American National Standards Institute*).
- **API:** siglas en inglés de Interfaz de Programación de Aplicaciones (*Application Programming Interface*), es un conjunto de especificaciones de comunicación entre componentes software y representa un método para conseguir abstracción en la programación.
- **ASP:** *Active Server Pages*, tecnología desarrollada por Microsoft para sus servidores de páginas Web.
- **Audiodescripción:** breve narración o descripción sonora mediante una “voz en *off*” de lo que ocurre en la pantalla en los momentos en los que hay silencio o pausa en el diálogo.
- **Base de datos:** de forma generalizada, una base de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior
- **BIF EC:** siglas en inglés de Confederación Europea de Lesionados Cerebrales y Familias (*Brain Injured & Families European Confederation*)
- **CERMI:** Comité Español de Representantes de Personas con Discapacidad, plataforma de representación, defensa y acción de los ciudadanos españoles con discapacidad.
- **CESyA:** Centro Español de Subtitulado y Audiodescripción.
- **CGI:** siglas en inglés de Pasarela de Interfaz Común (*Common Gateway Interface*) es una importante tecnología que permite a un cliente o navegador solicitar datos de un programa ejecutado en un servidor Web.
- **CSS:** siglas en inglés para Hojas de Estilo en Cascada (*Cascading Style Sheets*) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML.
- **CTIC:** Centro Tecnológico de la Información y la Comunicación.

- Daño cerebral: lesión que se produce en las estructuras cerebrales de forma súbita en personas nacidas sin ningún tipo de daño en el cerebro.
- EEUU: Estados Unidos
- FEDACE: Federación Española de Daño Cerebral.
- GANTT: popular herramienta gráfica fue desarrollada por Henry L. Gantt (1861-1919), cuyo objetivo es el de mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.
- HTML: siglas en inglés de Lenguaje de Marcación de Hipertexto (*Hyper Text Markup Language*) s un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.
- HTTP: siglas en inglés de Protocolo de Transferencia de Hipertexto (*Hypertext Transfer Protocol*), es un protocolo para transferir archivos o documentos hipertexto a través de la red.
- JSP: *Java Server Pages*, es la tecnología para generar páginas Web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en *scripts* que utilizan una variante del lenguaje java.
- PHP: *Hypertext Preprocessor*, es un lenguaje de programación interpretado. Se utiliza, entre otras cosas, para la programación de páginas Web activas y destaca por su capacidad de mezclarse con el código HTML.
- SABADO: Subtitulado y Audiodescrición, Base de Datos. Base de datos de CESyA sobre todo aquel material audiovisual que se encuentra subtitulado y/o audiodescrito en España.
- *Script*: programa o una secuencia de instrucciones que es interpretado y llevado a cabo por otro programa en lugar de ser procesado por el procesador de la computadora.
- *Servlet*: Un programa Java del lado del servidor que ofrece funciones suplementarias al servidor.
- SGBD: Sistema Gestor de Bases de Datos.
- SIDAR: Seminario Iberoamericano sobre Discapacidad y Accesibilidad en la Red.
- SQL: siglas en inglés de Lenguaje de Consulta Estructurado (*Structure Query Language*) lenguaje declarativo de acceso a bases de datos relacionales, que permite especificar diversos tipos de operaciones sobre las mismas.

- Subtitulado: proyección de texto escrito que describe los diálogos o sonidos de referencia que ocurren en las imágenes a las que acompañan.
- W3C: siglas en inglés del Consorcio *World Wide Web* (*World Wide Web Consortium*).
- WAI: siglas en inglés de Iniciativa de Accesibilidad Web (*Web Accessibility Initiative*).
- WBS: siglas de *Work Breakdown Structure*, diagrama jerarquizado con las tareas que constituyen el desarrollo de un proyecto.
- Web: Documento situado en la red informática de Internet, al que se accede mediante enlaces de hipertexto.

11 BIBLIOGRAFÍA

1.1 *Impresa*

[Despertar, 2005] Boletín de noticias de FEDACE. “Plan Director 2005-2008: un marco estratégico de acción para FEDACE”, diciembre 2005, página 10.

[Diseño de bases de datos relacionales, 1999] “Diseño de bases de datos relacionales”, escrito por Adoración de Miguel. Editorial Ra-Ma.

[Guía de recursos de atención al daño cerebral, 2005] Guía editada a finales de 2005 por FEDACE con motivo de la conmemoración “X Aniversario FEDACE, Esperanza-Experiencia”. En ella se recopila de forma escrita gran parte de los recursos disponibles en España sobre el tratamiento del daño cerebral adquirido.

[HTML 4.0, 1998] “HTML 4.0”, escrito por E. Stephen Mack y Janan Platt. Editorial Anaya Multimedia, colección Biblioteca Profesional.

[Usabilidad: diseño de sitios Web, 2002] “Usabilidad: diseño de sitios Web”, escrito por Jacob Nielsen. Editorial Prentice Hall.

[Diseño de páginas Web, 1997] “Diseño de páginas Web: Técnicas avanzadas” escrito por David Siegel y editado por Anaya Multimedia, Diseño y Creatividad.

[Mejorar tus páginas Web, 1998] “Mejorar tus páginas Web”, escrito por Elisabeth A. Parker y editado por Anaya Multimedia.

1.2 En formato electrónico

- [1] Internet Valley WWW Links Community: History of the Internet and WWW
<http://netvalley.com/history-refer.html>
- [2] World Wide Web Consortium (W3C), Guía breve de accesibilidad
<http://www.w3c.es/divulgacion/guiasbreves/Accesibilidad>
- [3] Web Accessibility Initiative (WAI)
<http://www.w3.org/WAI/>
- [4] Haces Falta
<http://www.hacesfalta.org/>
- [5] Federación Española de Daño Cerebral
<http://www.fedace.org/web/>
- [6] Centro Español de Subtitulado y Audiodescripción (CESyA)
<http://www.cesya.es>
- [7] Universidad Carlos III de Madrid, Servicio de Comunicación e Imagen Institucional
<http://www.uc3m.es/noticiasIII/>
- [8] Estatutos de la Universidad Carlos III de Madrid.
<http://www.uc3m.es/uc3m/gral/UH/estatutos-2002.html>
- [9] Desarrollo Web
<http://www.desarrolloweb.com/>
- [10] PHP: Hypertext Preprocessor
<http://www.php.net/>
- [11] Wikipedia, la enciclopedia libre
<http://es.wikipedia.org/wiki/Portada>
- [12] Apuntes de PHP: Funciones de usuario
<http://www.ignside.net/man/php/functions.php>
- [13] Web Experto
<http://www.webexperto.com>
- [14] HTML Point, el portal de la publicación Web
http://www.htmlpoint.com/perl/perl_02.htm
- [15] Microsoft SQL Server Home
<http://www.microsoft.com/sql/default.mspx>

- [16] Oracle Corporation
<http://www.oracle.com/index.html>
- [17] Firebird, Relational Data Base for the New Millennium
<http://firebird.sourceforge.net/>
- [18] MySQL AB: The world's most popular open source database
<http://www.mysql.com/>
- [19] PostgreSQL
<http://www.postgresql.org>
- [20] SQLite Home Page
<http://www.sqlite.org/>
- [21] Apache Software Foundation
<http://www.apache.org/>
- [22] Netcraft, empresa de servicios de seguridad en Internet, rastreo y estadísticas Web
http://news.netcraft.com/archives/2003/01/01/about_netcraft.html
- [23] Web del Congreso Nacional de Software Libre 2006 (México). Proyecto Cherokee: Diseño, implementación y aspectos de rendimiento de servidores Web.
<http://www.consol.org.mx/>
- [24] Cherokee Web Server
<http://www.0x50.org/>
- [25] Noticia de la edición digital del periódico El País, 5 de enero de 2006
http://www.elpais.es/articulo/elpportec/20060105elpcibpor_2/Tes
- [26] World Wide Web Consortium
<http://www.w3c.es>
- [27] Pautas de Accesibilidad al Contenido en la Web
<http://www.w3.org/TR/WCAG10/>
- [28] Test de Accesibilidad Web
<http://www.tawdis.net>
- [29] HERA, revisando la accesibilidad con estilo
<http://www.sidar.org/hera/>
- [30] Discapnet, el portal de la discapacidad
<http://www.discapnet.es>
- [31] Accesoweb, lista en castellano sobre problemas y soluciones de diseño accesible para la Red
<http://es.groups.yahoo.com/group/accesoweb/>

- [32] La Web del voluntariado
<http://www.voluntariado.net>
- [33] Pautas de Accesibilidad para Herramientas de Autor
<http://www.w3.org/TR/ATAG10/>
- [34] Pautas de Accesibilidad para XML
<http://www.w3.org/TR/xag>
- [35] Pautas de Accesibilidad para Agentes de Usuario
<http://www.w3.org/TR/UAAG10/>
- [36] Técnicas para Pautas de Accesibilidad al Contenido en la Web
<http://www.w3.org/TR/WCAG10-TECHS/>
- [37] DynDNS, obtención gratuita de dominios y otros servicios asociados
<http://www.dyndns.com/>
- [38] Técnicas Fundamentales para las Pautas de Accesibilidad al Contenido en la Web 1.0
<http://www.w3.org/TR/WCAG10-CORE-TECHS/>
- [39] Técnicas CSS para las Pautas de Accesibilidad al Contenido en la Web 1.0
<http://www.w3.org/TR/WCAG10-CSS-TECHS/>
- [40] Técnicas HTML para las Pautas de Accesibilidad al Contenido de la Web 1.0
<http://www.w3.org/TR/WCAG10-HTML-TECHS/>
- [41] GrayBit 1.0: Grayscale Conversión Contrast Accessibility Tool
<http://graybit.com>
- [42] The W3C Markup Validation Service
<http://validator.w3.org/>
- [43] Fundación CTIC
<http://www.fundacionctic.org/>
- [44] Examiner, revisa y evalúa la accesibilidad en la Web
<http://www.accesible.com.ar/examinator/>
- [45] HTML 4.01 Specification
<http://www.w3.org/TR/REC-html40>
- [46] InfoJobs, bolsa de empleo y ofertas de trabajo
<http://www.infojobs.net/>

ANEXO A: CUESTIONARIO SOBRE LA BASE DE DATOS

Cuestiones generales:

1. ¿Cuál es el propósito de la asociación?

2. ¿Qué tipo de servicio presta a sus miembros?

3. ¿Por qué se considera necesario una base de datos?

4. ¿Quién trabajará con la base de datos? A todos los niveles.

Cuestiones generales para cada tipo de usuario (administra y/o consulta):

5. ¿Cuál es su cometido en su puesto de trabajo?

--

6. ¿Qué herramientas software usa principalmente para ello?

--

7. ¿Cuáles son sus conocimientos sobre bases de datos?

a. Diseño	
b. Administración	
c. Consultas	

8. ¿Con qué tipo de datos trabaja?

--

9. ¿Qué tipo de informes utiliza?

--

Cuestiones sobre los datos a almacenar:

- 10.** A grandes rasgos, ¿cuál es la información que se desea almacenar en la base de datos? (*Identificación de entidades por parte del analista*)

- 11.** ¿Qué información necesita almacenar sobre cada elemento (*entidad*) perteneciente a la base de datos? (*En función de las identificadas anteriormente*)

Nombre	Tipo	Opcional	Repetitivo	Compuesto	Comentarios

Nombre	Tipo	Opcional	Repetitivo	Compuesto	Comentarios

12. ¿Qué tipo de información adicional consideraría útil a partir de ahora?

Cuestiones sobre los requisitos del sistema:

13. ¿Qué operaciones se realizarán en la base de datos (consultas, inserciones, modificaciones)?

14. ¿Cuándo se realizarán éstas operaciones? Por ejemplo, una vez al mes, a la semana, a diario en horario laboral, a diario en cualquier momento, etc.

15. ¿Desde dónde se accederá? Por ejemplo, desde un único terminal, desde equipos conectados a una intranet, vía web, etc.

16. ¿Quién estará autorizado a insertar/modificar datos?

17. ¿Quién estará autorizado a consultar la base de datos?

18. ¿Se desea almacenar datos históricos de algún tipo?

19. ¿Considera necesario hacer copias de seguridad? En caso afirmativo, ¿con qué frecuencia?

20. ¿Cuál será el tamaño aproximado inicial de la base de datos? Número de tuplas.

21. ¿Qué tasa de crecimiento tendrá? Inserciones diarias, semanales, esporádicas, etc.

22. ¿Sobre que campos considera que deberán hacerse las consultas?

23. ¿Generación de algún tipo de informe?

24. ¿Optimización de tiempo y/o espacio?

25. ¿Existirá comunicación con alguna herramienta software?

26. ¿Acceso vía Internet?

27. ¿De qué recursos hardware y software dispone?

a. Equipos informáticos

	Procesador	Memoria RAM	Capacidad HD	Sistema Operativo
1				
2				
3				
4				
5				

b. Otros componentes (impresoras, escáner)	
c. SGBD (comerciales, gratuitos)	
d. Intranet.	
e. Conexión a Internet (ancho de banda, proveedor)	
f. Servidor.	

Cuestiones sobre la actual base de datos (en caso de existir):

28. ¿Sobre qué SGBDs o formato se encuentra la actual base de datos?

29. ¿Cuál es su tamaño aproximado (*bytes*)?

30. ¿Existe alguna aplicación que use la actual base de datos?

ANEXO B: CUESTIONARIO SOBRE LA APLICACIÓN WEB

Cuestiones sobre el usuario/administrador:

1. ¿Es usuario habitual de Internet?(“sí” continuar por la pregunta 2, “no” pasar a la pregunta 5)

2. ¿Con qué frecuencia navega?

3. Motivos: trabajo, estudios, ocio, etc.

4. ¿Qué navegador/es y versión utiliza habitualmente?

Cuestiones sobre la apariencia de la página:

5. ¿Desea que la página sea cargada en la ventana actual de trabajo o que se abra en una ventana nueva?

6. ¿Qué tamaño desea para la ventana?

7. ¿Preferencia de visualización con algún navegador en concreto?

8. ¿Qué vínculos destacados le gustaría que estuvieran siempre presentes? Por ejemplo, página principal, información corporativa, correo electrónico de contacto o sugerencia, enlaces externos, etc.

9. ¿Seguir alguna hoja de estilo existente? (“sí” pasar a la pregunta 14, “no” continuar por la pregunta 10)

10. ¿Fondo de color o con imagen?

11. ¿Imagen corporativa presente?

12. Con respecto al formato del texto:

a. Tipo de fuente.	
b. Tamaño.	
c. Color.	

- 13.** ¿Debe prestarse especial atención a posibles usuarios con alguna discapacidad?
(enumerar las posibles discapacidades en concreto, si es posible)

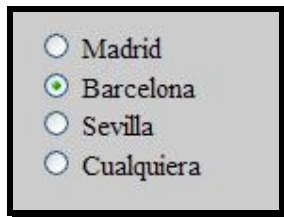
--

Cuestiones sobre el formulario de consulta:

- 14.** ¿Cada resultado de una consulta debe aparecer en la actual ventana de trabajo o en una nueva ventana?

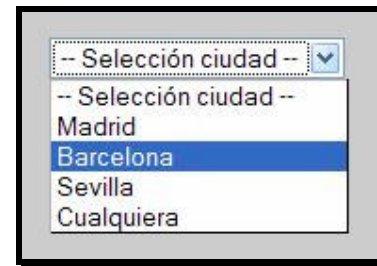
- 15.** Selección de tipos enumerados (pertenecen a un conjunto restringido y sólo ha de elegir uno para la búsqueda)

a. Casillas excluyentes.



☐ Madrid
☒ Barcelona
☐ Sevilla
☐ Cualquiera

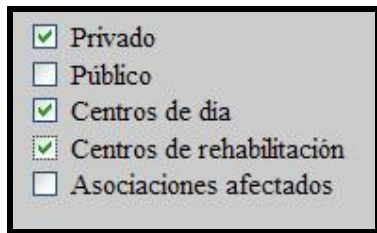
b. Menú desplegable.




-- Selección ciudad --
-- Selección ciudad --
Madrid
Barcelona
Sevilla
Cualquiera

- 16.** Selección combinada.

a. Casillas de verificación.



☒ Privado
☐ Público
☒ Centros de día
☒ Centros de rehabilitación
☐ Asociaciones afectados



-- Seleccione criterio --
-- Seleccione criterio --
-- Seleccione criterio --
-- Seleccione criterio --
-- Seleccione criterio --

b. Varios menús desplegables.

17. Cuadros de texto.

a. Tipo de letra.	
b. Color del texto.	

Nombre	<input type="text"/>
Apellidos	<input type="text"/>

ANEXO C: INSTALACIÓN DE LA BASE DE DATOS Y DE LA APLICACIÓN WEB DE FEDACE

Este capítulo pretende servir de guía a la hora de instalar todos los elementos necesarios para la puesta en marcha y funcionamiento de la BD. Los pasos están expresados en un lenguaje lo menos técnico posible y sin dar por sentado grandes conocimientos informáticos ya que su objetivo es poder ser entendido por cualquier persona, independientemente de sus conocimientos informáticos, cargo o función. Todas las aplicaciones aquí mencionadas y otros documentos de ayuda se encuentran en el CD adjunto a este documento.

1. Requisitos previos

Los siguientes pasos de instalación suponen que se está trabajando sobre un sistema operativo de la familia Windows XP, preferiblemente de reciente instalación. La instalación del sistema debería ser igualmente similar en cualquier sistema operativo Windows 2000 o Windows NT.

El equipo informático debe contar con un procesador mínimo de 1,5 ó 2 MHz, memoria RAM mínima de 256 MB y con una memoria libre en disco duro de aproximadamente 200 MB.

Las instrucciones de instalación han sido redactadas en un lenguaje lo menos técnico posible y se han acompañado de numerosas capturas de pantalla para que sea más fácil seguir cada uno de los pasos. Aún así, se sobreentiende que la persona que va a llevar a cabo la instalación del sistema tiene alguna experiencia, aunque sea escasa, en la instalación de algún tipo de software o incluso en las tareas habituales de mantenimiento y explotación de cualquier PC doméstico o de oficina.

2. Instalar MySQL

MySQL es un Sistema Gestor de Bases de Datos (SGBD). Es decir, es un programa software que gestiona todo lo relacionado con la base de datos. Para poder alojar una base de datos es necesario siempre instalar previamente un SGBD que se encargue de gestionarla y nos permite, entre otras cosas, crearla.

En el CD de instalación encontrará una carpeta llamada “MySQL” y dentro de ella otra carpeta que contiene el archivo instalable de MySQL. La versión que se va a instalar de MySQL es la 5.0.15, puede encontrarla también para descargar en la página oficial de MySQL www.mysql.com. Pinche sobre este archivo y ejecútelo.

Siga los siguientes pasos para una correcta instalación:

1. En la primera pantalla pulse “*Next*”.

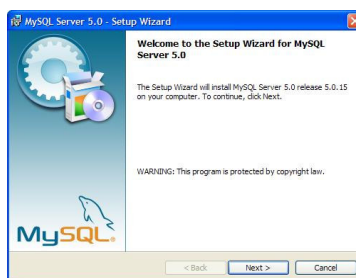


Figura C-1 Instalación de MySQL. Paso 1

2. Elija a continuación la opción “*Typical*”. Pulse “*Next*”.



Figura C-2 Instalación de MySQL. Paso 2

3. Pulse ahora “*Install*”. El programa se instalará por defecto en “C:\Archivos de programa\MySQL”

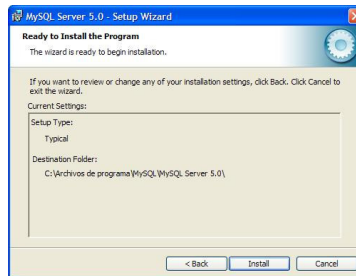


Figura C-3 Instalación de MySQL. Paso 3

4. La instalación durará unos segundos.

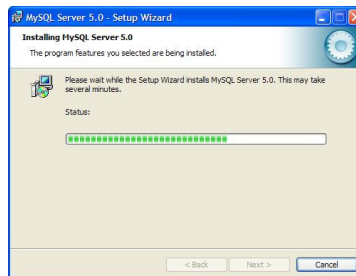


Figura C-4 Instalación de MySQL. Paso 4

5. Marque la opción “*Skip Sign-Up*” y pulse “*Next*”.



Figura C-5 Instalación de MySQL. Paso 5

6. Seleccione la opción “*Configure the MySQL Server now*” y pulse “*Finish*”.



Figura C-6 Instalación de MySQL. Paso 6

7. En la siguiente pantalla pulse “*Next*”.

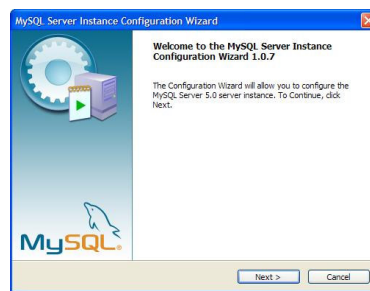


Figura C-7 Instalación de MySQL. Paso 7

8. Elija la opción “*Detailed Configuration*” para llevar a cabo una configuración detallada. Pulse “*Next*”.



Figura C-8 Instalación de MySQL. Paso 8

9. Seleccione la opción “*Server Machine*” y pulse “*Next*”.

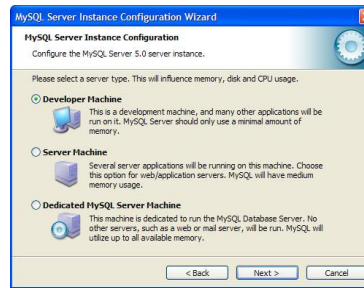


Figura C-9 Instalación de MySQL. Paso 9

10. Seleccione la opción “*Transactional Database Only*”. Pulse “*Next*”.



Figura C-10 Instalación de MySQL. Paso 10

11. En la siguiente pantalla no cambie nada y pulse “*Next*”.

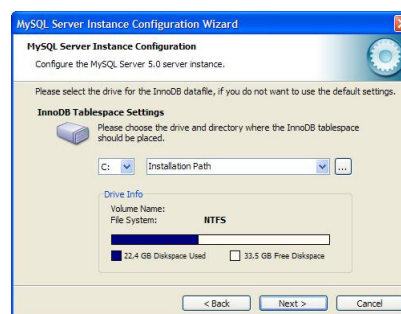


Figura C-11 Instalación de MySQL. Paso 11

12. Marque la opción “*Decision Support (DSS)/OLAP*” y pulse “*Next*”.

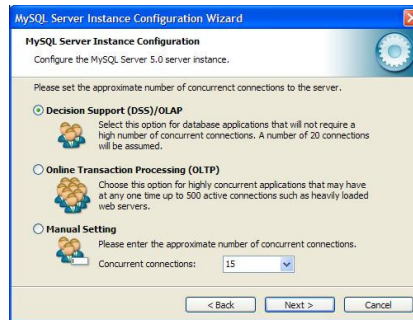


Figura C-12 Instalación de MySQL. Paso 12

13. En la siguiente pantalla, asegúrese de que las opciones “*Enable TCP/IP Networking*” y “*Enable Strict Mode*” estén seleccionadas. Además, donde pone “*Port Number*” debe estar introducido el valor 3306. Pulse “*Next*”.

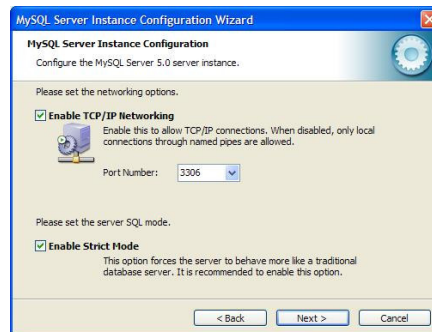


Figura C-13 Instalación de MySQL. Paso 13

14. Seleccione la opción “*Standard Character Set*”. Pulse “*Next*”.



Figura C-14 Instalación de MySQL. Paso 14

15. Marque las opciones “*Install As Windows Service*”, “*Launch the MySQL Server automatically*” y “*Include Bin Directory in Windows PATH*”. Pulse “*Next*”.



Figura C-15 Instalación de MySQL. Paso 15

16. En la siguiente pantalla marque solo la opción “*Modif. Security Settings*” e introduzca dos veces la palabra “*fedace.dca*” en minúsculas. Ésa será la contraseña de acceso a MySQL. Pulse “*Next*”.



Figura C-16 Instalación de MySQL. Paso 16

17. Para terminar, pulse el botón “*Execute*”.

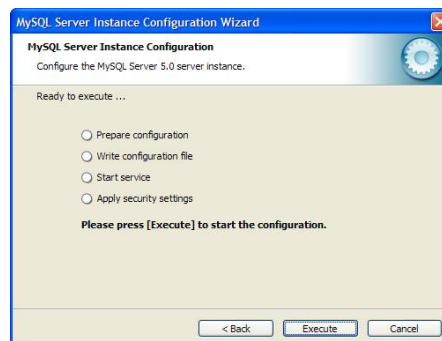


Figura C-17 Instalación de MySQL. Paso 17

18. Salga pulsando el botón “*Finish*”.



Figura C-18 Instalación de MySQL. Paso 18

3. Instalar el servidor Apache

Para poder alojar la aplicación Web en un equipo y proporcionar acceso a ella desde Internet, es necesario instalar un programa software específico denominado servidor, que hace de intermediario entre la aplicación Web y el usuario que desea hacer uso de ella de forma remota.

La versión a instalar del servidor Web será Apache 2.0.55. En el contenido del CD de instalación podrá encontrar una carpeta denominada “Apache Server” y dentro de ella el archivo instalable de la aplicación. Igualmente, desde la página de “*The Apache Software Foundation*” dedicada al servidor Apache, <http://httpd.apache.org/>, se puede descargar de forma gratuita el programa. Concretamente se buscará por el archivo `apache_2-1.0.55-win32-x86-no_ssl`, que es el instalador para sistemas operativos Windows.

Una vez descargado o localizado en el CD, se ha de ejecutar. Durante el proceso de instalación, además de pedir que se acepten los términos legales, preguntará el nombre del dominio y del servidor, que puede rellenar como “localhost” en ambos casos. Solicitará también una dirección de correo electrónica, que será la del administrador o persona que vaya a mantener el servidor. A esta dirección se remitirán los posibles problemas que tengan los usuarios de las páginas alojadas en el servidor. Por último preguntará si desea que el servidor responda por el puerto 80 para todos los usuarios o si

desea que sólo se active para el usuario actual en el puerto 8080. Seleccione la primera opción (“*for all users, on port 80, as a service*”).

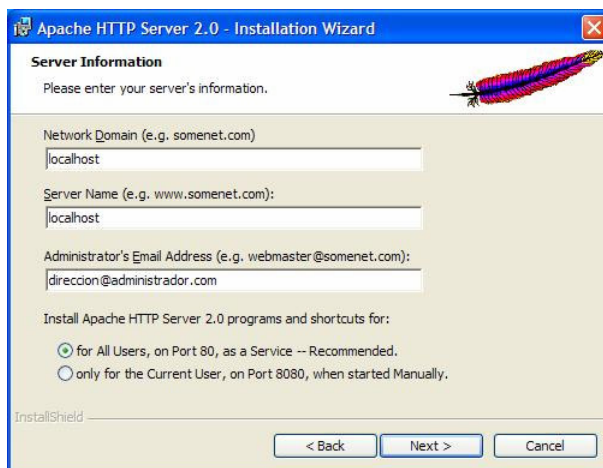


Figura C-19 Instalación de Apache

Especificaremos la instalación típica (“*Typical*”) y como destino la que especifique por defecto “C:/Archivos de programa/Apache Group”. Después de pulsar el botón “*Install*” el programa tardará unos minutos en instalarse. Al terminar, se nos dirá que todo ha finalizado exitosamente.

Atención: Es muy recomendable instalar el servidor Apache en una instalación Windows limpia, antes que otro tipo de programas sobre todo si éstos hacen uso de Internet. Un error muy común es el que se produce cuando hay otro programa “escuchando” por el puerto 80 (*make_sock: could not bind to address 0.0.0.0:80*), generalmente otro programa servidor diferente a Apache o aplicaciones como *Skype* o similares. En ese caso, las aplicaciones deben ser apagadas, reconfiguradas o incluso desinstaladas. En Windows XP es probable que esté instalado *Internet Information Server*, un servidor de Microsoft. Para quitarlo, vaya a “Inicio → Panel de Control → Agregar o Quitar Programas” y seleccione el botón “Agregar o quitar componentes de Windows”. Aparecerá una lista con varios programas y servicios. Asegúrese de que la casilla referente a *Internet Information Server* está desactivada. Si lo estaba, al desactivarla y darle a “Siguiente” se desinstalará de su equipo.

Para comprobar el correcto funcionamiento del servidor, basta con abrir una ventana de cualquier navegador y visitar la dirección `http://localhost/`. Si aparece una página de bienvenida, Apache está correctamente instalado en su equipo.

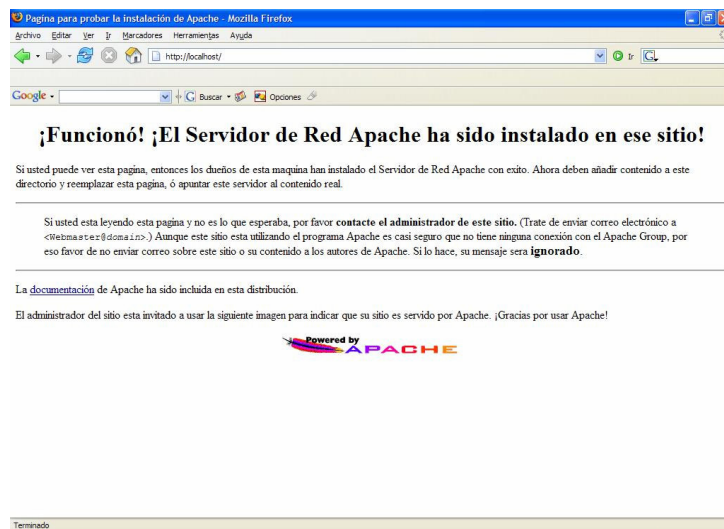


Figura C-20 Página de bienvenida de Apache

Para cualquier tipo de consulta o ayuda adicional, por favor, consulte la página de *The Apache Software Foundation* dedicada al servidor Apache, `http://httpd.apache.org/`. Allí podrá encontrar manuales y preguntas frecuentes sobre el programa.

4. Configurar el servidor Apache

El servidor Apache, una vez instalado, tiene por defecto configuradas unas directivas particulares de seguridad. Esta configuración no es suficiente y es necesario adaptarla para que contemple una mayor seguridad y evite accesos malintencionados a la aplicación Web o a otros recursos desde el servidor.

En el CD de instalación encontrará un archivo llamado `httpd.conf` que no es más que el archivo de configuración del servidor Apache modificado para ofrecer una mayor protección frente a intrusos. Copie este archivo y péguelo en la carpeta “C:\Archivos de programa\Apache Group\Apache2\conf”, reemplazando al que ya existe en esta carpeta

con ese mismo nombre. Reinicie a continuación el ordenador. Si no ha instalado Apache en el directorio por defecto, deberá cambiar en este archivo las rutas por las correctas.

5. Instalar el módulo de PHP

El lenguaje PHP (*Hypertext Preprocessor*) es ampliamente utilizado en la red para implementar páginas dinámicas. Este tipo de lenguaje se denomina “del lado del servidor” puesto que el navegador que utiliza el usuario no es capaz de interpretarlo por sí mismo y necesita que el servidor lo “traduzca” a un tipo de página que sí sepa como interpretar (código HTML). Es por ello que si tenemos un servidor instalado, será sumamente interesante disponer también de una configuración de PHP para poder ofrecer páginas escritas en este lenguaje.

La siguiente instalación se hará con la versión 5 de PHP, sobre un equipo con sistema operativo Windows XP Professional y que tiene el servidor Apache 2.0 instalado y correctamente funcionando.

Encontrará los archivos necesarios para la instalación en el CD adjunto a este manual, concretamente en la carpeta de nombre “PHP” encontrará un archivo comprimido llamado `php-5.1.1-Win32.zip`. También lo que se puede hacer es obtener los archivos para empezar la instalación en la página oficial, <http://www.php.net/>, y buscar la versión 5 para descargar, se ha de localizar el archivo llamado `php-5.1.1-Win32.zip`. Existen dos formas de instalar PHP en un equipo: como CGI o como módulo de Apache. En nuestro caso, será como módulo del servidor Apache por ser la más recomendada y más segura.

Una vez que tiene localizado los archivos necesarios, siga los siguientes pasos:

1. Descomprima los archivos en el directorio recomendado “C:\Php”.
2. Entre en la carpeta recién creada “C:\Php”, copie el archivo llamado `php_mysql.dll` que está ubicado en “C:\Php\ext” y péguelo en “C:\Php”.
3. Busque en esta misma carpeta, “C:\Php”, un archivo llamado `php5ts.dll`. Cópielo y péguelo en la carpeta “C:\Windows\System”.

4. En la carpeta “C:\Php”, de nuevo, existe un archivo llamado `php.ini-recommended`. Debe copiarlo y pegarlo en la carpeta “C:\Windows”.
5. Cambie el nombre al archivo `php.ini-recommended` y póngale simplemente `php.ini`. Comprobará que el icono con el que está representado el archivo cambia.
6. A continuación, debe abrir este archivo `php.ini` y buscar las siguientes expresiones:
 - a. Busque `extension='php_mysql.dll'` y asegúrese de que no hay ningún punto y coma “;” delante de esta expresión. En tal caso, quítelo.
 - b. Busque `doc_root` y escriba la dirección de la carpeta “htdocs” del servidor Apache. Por ejemplo, si instaló Apache en el directorio raíz C, la dirección a poner aquí será `doc_root='C:\Archivos de programa\Apache Group\Apache2\htdocs'`.
 - c. Busque `extension_dir` y ponga `extension_dir='C:\Php'`.
 - d. Busque `arg_separator.output="& " y quítele el punto y coma “;” que tiene delante.`
 - e. Busque `magic_quotes_gpc = Off` y ponga `magic_quotes_gpc = On`
 - f. Guarde todos los cambios.
7. Vaya a la carpeta de configuración de Apache, por ejemplo, `C:\Archivos de programa\Apache Group\Apache2\conf` y abra el archivo `http.conf`. Allí debe copiar exactamente, al final del archivo, las siguientes líneas:

```
LoadModule php5_module "c:/php/php5apache2.dll"  
AddType application/x-httpd-php .php  
PHPIniDir "C:/Windows"
```

Tenga presente siempre que, si instaló Apache o descomprimió PHP en otras carpetas diferentes a las aquí recomendadas, deberá escribir en su lugar la ruta correcta en todos los pasos anteriormente explicados.

Ahora compruebe que todo funciona correctamente. Para asegurarse de que su servidor Apache es capaz de interpretar páginas Web escritas en lenguaje PHP, nada más fácil

que crear una sencilla página en PHP e intentar visualizarla. Reinicie primero el equipo y a continuación haga lo siguiente:

- Abra el Bloc de Notas o Notepad.
- Escriba el siguiente texto: `<?php phpinfo() ?>`
- Seleccione “Archivo → Guardar como” y llámelo `prueba.php`
- Por último, copie el archivo que acaba de crear y pégelo en “C:\Archivos de programa\Apache Grupo\Apache2\htdocs”

Abra el navegador que habitualmente use y escriba en la barra de dirección `http://localhost/prueba.php`. Si es capaz de visualizar la página mostrada abajo, PHP se ha instalado con éxito en su equipo y su servidor podrá ofrecer páginas escritas en éste lenguaje.

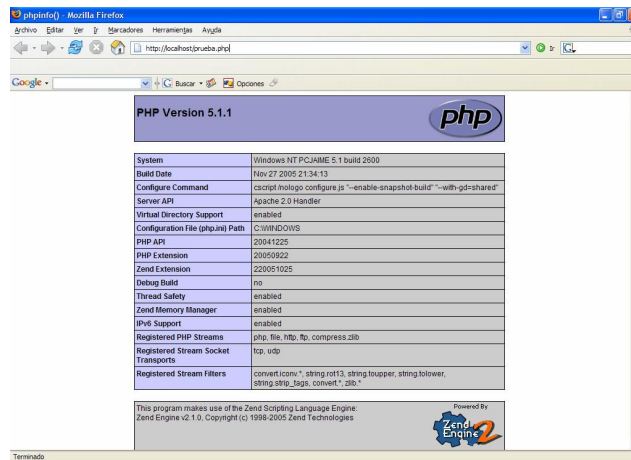


Figura C-21 Página mostrada por `phpinfo()`

Para más información, puede visitar la página oficial de php, www.php.net.

6. Instalar phpMyAdmin

Para gestionar de forma más fácil y cómoda las bases de datos creadas sobre MySQL, existe una aplicación Web gráfica llamada phpMyAdmin que simplifica enormemente el trabajo. En este caso, no será necesario manipular directamente la base de datos creada en MySQL puesto que todas las posibles tareas de administración se pueden hacer desde la aplicación Web diseñada para ello. Sin embargo, phpMyAdmin es de gran utilidad para realizar, por ejemplo, copias de seguridad del contenido de la base de datos o cargas iniciales en el sistema. Para instalar phpMyAdmin siga los siguientes pasos:

8. En el CD de instalación encontrará una carpeta llamada “phpMyAdmin” y dentro un archivo llamado `phpMyAdmin-2.7.0-pl2.zip`. También puede buscar este archivo en la página principal de esta aplicación en Internet, www.phpmyadmin.net.
9. Descomprímalo en la carpeta “C:\Archivos de programa\Apache Group\Apache2\htdocs”.
10. Cambie el nombre de la carpeta recién creada a phpMyAdmin y entre en ella.
11. Debe crear, con el Bloc de Notas o NotePad por ejemplo, un archivo llamado `config.inc.php`.
12. Abra el archivo recién creado y escriba, literalmente, lo siguiente:

```
<?php
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin/';
$cfg['blowfish_secret'] = 'basededatosfedace';
//Cualquier semilla para la clave
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['auth_type'] = 'cookie';
$cfg['Servers'][$i]['user'] = 'root'; // Usuario
$cfg['Servers'][$i]['password'] = 'fedace.dca'; //
Contraseña
$cfg['Servers'][$i]['only_db'] = 'fedace'; // Nombre de
la BD
```

?>

13. Guarde el archivo con los cambios realizados.

Para acceder a esta aplicación, debe teclear en el navegador la dirección `http://localhost/phpMyAdmin/index.php`. Es muy recomendable que la guarde en “Favoritos” para poder acceder a ella más fácilmente.

7. Cargar la base de datos de FEDACE en MySQL

En el CD de instalación podrá encontrar un archivo llamado `fedace.sql`. Este archivo contiene toda la estructura de la base de datos que sustenta la Guía de Recursos sobre Daño Cerebral Adquirido y es necesario cargarla en el gestor de bases de datos MySQL. La carga se realizará, por su sencillez y rapidez, desde la aplicación phpMyAdmin siguiendo los siguientes pasos:

1. Entre en phpMyAdmin desde el navegador (recuerde que la dirección, si ha seguido los pasos de instalación mostrados en la sección 3.6 *Instalar phpMyAdmin* es <http://localhost/phpMyAdmin/index.php>).

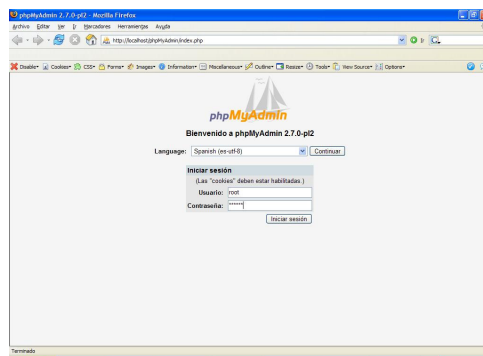


Figura C-22 Carga de la base de datos. Paso 1

2. Elija la opción que dice “Importar”.

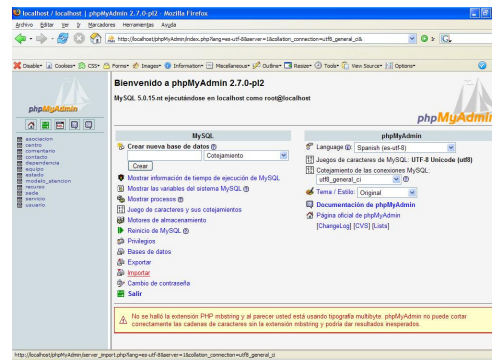


Figura C-23 Carga de la base de datos. Paso 2

3. Pulse el botón que dice “Examinar” y localice la ubicación en el CD de instalación del fichero de carga de la base de datos `fedace.sql`.

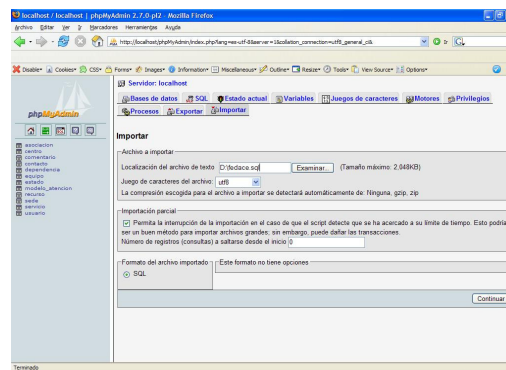


Figura C-24 Carga de la base de datos. Paso 3

4. Pulse “Continuar”.
5. Si todo ha ido correctamente, se le indicará mediante un mensaje de confirmación.

8. Instalar la aplicación Web

A continuación, y ya para finalizar la instalación de todo el sistema, se debe instalar la aplicación Web de la Guía de Recursos sobre Daño Cerebral Adquirido en el servidor Web Apache previamente instalado en el equipo. Para ello, se deben seguir los siguientes pasos:

1. Localice en el CD de instalación una carpeta llamada `bdfedace`.
2. Entre en la carpeta y seleccione todos sus archivos (Edición -> Seleccionar todo) y cópielos.
3. Pegue todos los archivos en la carpeta “C:\Archivos de programa\Apache Group\Apache2\htdocs”.

9. Obtener un dominio gratuito en Internet

Actualmente existen en Internet varias empresas que proporcionan dominios gratuitos a cambio de alojar publicidad o incluso sin ella. Un dominio es una dirección IP que tiene asociada un nombre fácil de recordar y de entender y que permite acceder a un servidor que aloja servicios Web o archivos descargables, entre otros.

Para la instalación de esta Guía de Recursos sobre Daño Cerebral Adquirido se ha optado por la empresa DynDns, dado su carácter absolutamente gratuito, su fácil funcionamiento y que no exige ofrecer publicidad a cambio. La dirección en Internet de esta empresa es www.dyndns.com.

9.1 Registro y selección del dominio

Una vez dentro de la página de DynDns el primer paso es registrarse mediante un nombre de usuario y una contraseña. Se debe intentar localizar un enlace que indique algo similar a “*Sign Up Now*” y pinchar en él.

Después de aceptar los términos y condiciones bajo los cuales se contrata el servicio gratuito con DynDns, la información que se nos pide es:

- un nombre de usuario
- una clave de acceso
- un correo electrónico

Una vez elegidos estos datos, se debe pulsar el botón “*Continue*” o similar y esperar a recibir en la dirección de correo proporcionada un mensaje que comunique que se ha dado de alta exitosamente. A partir de entonces ya se puede proceder a identificarse en la página de inicio.

Al introducir el nombre de usuario y claves escogido, se permite contratar alguno de los servicios que se ofrece desde esta empresa. El servicio que se ha de solicitar a continuación es algo similar a “*Dynamic DNS*”. El siguiente paso simplemente es elegir el nombre que más guste para establecerlo como dominio personal.

9.2 Asociar la IP dinámica

Además de tener un dominio propio donde poder alojar las Web, hace falta además un programa que asocie de forma automática nuestra dirección IP dinámica al dominio que se acaba de adquirir. Cada ordenador que está conectado a Internet tiene una IP propia, que no es más que una secuencia de hasta 12 dígitos agrupados de 3 en 3 que identifica de manera única a ese ordenador. Las IP hoy día generalmente son dinámicas, esto es, que cada vez que el ordenador se conecta a Internet se le asigna una IP distinta a la anterior conexión.

Mediante la aplicación *DynDns Update*, de forma automática, se asocia nuestra dirección IP actual al nombre del dominio que tengamos dado de alta en DynDns, permitiendo que siempre que alguien visite nuestro dominio se redirija de forma automática a nuestra IP para que podamos ofrecer la Web correspondiente.

Puede conseguir este programa en la propia página de DynDns u obtenerlo directamente de CD de instalación. Es un archivo localizado dentro de la carpeta “DynDNS Update” llamado `dyndns-setup.zip`. Para instalarlo únicamente tiene

que descomprimir este archivo y ejecutar un archivo llamado `setup.exe`. La instalación es prácticamente automática.

Cuando haya terminado la instalación, sólo queda un último paso. Abra el programa recién instalado y pulse en la pestaña que dice “*Options*”. Allí, debe pulsar un botón que dice “*Install Service*”, lo que permitirá que cada vez que encienda el ordenador, el programa se pondrá en funcionamiento de forma automática y actualizará la IP que tenga en ese momento el equipo con el dominio registrado.

ANEXO D: COPIAS DE SEGURIDAD DE LA BASE DE DATOS

Es altamente recomendable realizar cada cierto tiempo copias de seguridad del contenido de la base de datos, para evitar pérdidas de los datos contenidos en ella. Las copias de seguridad deberían ser tenidas en cuenta especialmente cuando los datos sean numerosos o ante instalaciones u operaciones que pudieran ser críticas en el equipo informático donde está instalada la base de datos. Forma parte de las tareas del administrador decidir cuándo estas copias de seguridad deben tener lugar.

1 Hacer una copia de seguridad

Para hacer una copia de seguridad de la base de datos, siga el siguiente procedimiento:

1. Entre en phpMyAdmin desde el navegador (recuerde que la dirección, si ha seguido los pasos de instalación mostrados en la sección 2.5 *Instalar phpMyAdmin* es <http://localhost/phpMyAdmin/index.php>).

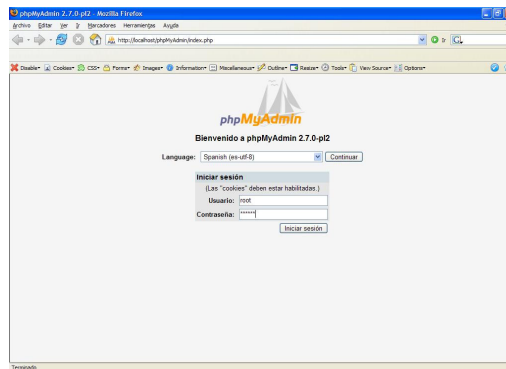


Figura D 1 Hacer copia de seguridad. Paso 1

2. Seleccione la opción “Exportar”

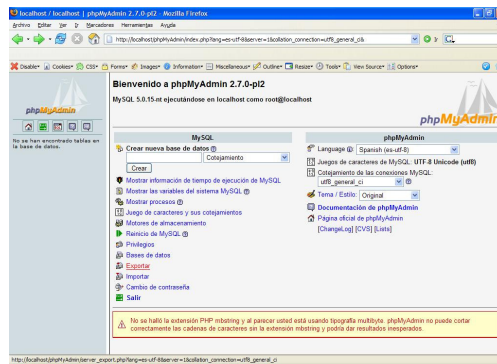


Figura D-2 Hacer copia de seguridad. Paso 2

3. En la siguiente pantalla, debe realizar las siguientes selecciones:

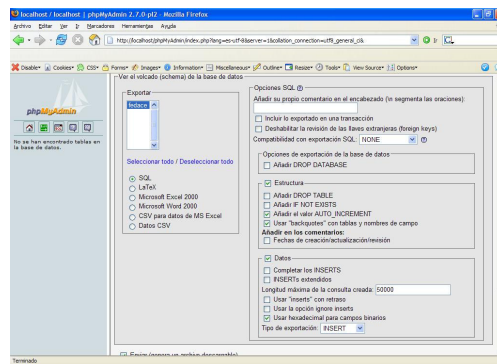


Figura D-3 Hacer copia de seguridad. Paso 3

- Dentro de la agrupación de opciones llamada “Exportar”, en la parte izquierda de la pantalla, seleccione la base de datos “fedace” y marque la opción “SQL”
- En la agrupación de opciones llamada “Opcione SQL”, en la parte derecha de la pantalla, marque la opción denominada “Añadir DROP DATABASE” y desmarque la opción “Usar ‘backquotes’ con tablas y nombres de campos”
- En la parte inferior de la pantalla, fuera de este último grupo de opciones, marque la opción “Enviar (genera un archivo descargable)” y en “Plantilla del nombre del archivo” escriba un nombre para el fichero donde se guardará la copia de la base de datos.

4. Pulse “Continuar”
5. El fichero con la copia de seguridad se habrá generado y guardado en la carpeta por defecto en la que habitualmente descarga sus archivos de Internet.

2 Recuperar una copia de seguridad

Para recuperar una copia de seguridad anteriormente hecha de la base de datos, se deben seguir los siguientes pasos:

1. Entre en phpMyAdmin desde el navegador (recuerde que la dirección, si ha seguido los pasos de instalación mostrados en la sección 2.5 *Instalar phpMyAdmin* es <http://localhost/phpMyAdmin/index.php>).

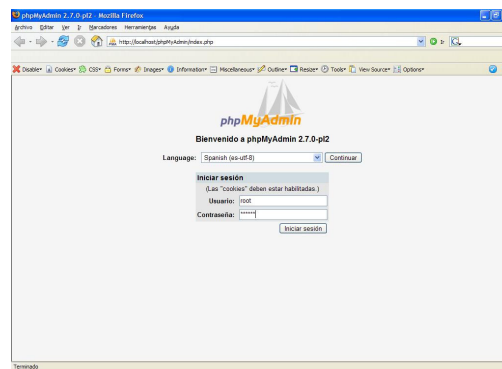


Figura D-4 Recuperar copia de seguridad. Paso 1

2. Pulse en la opción “Importar”.

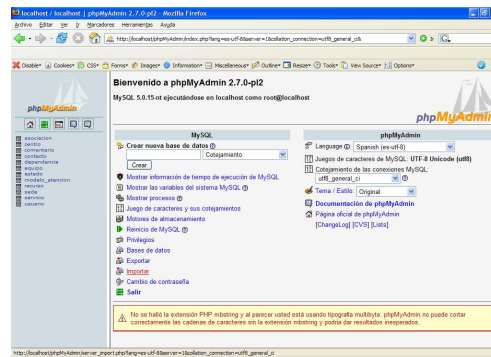


Figura D-5 Recuperar copia de seguridad. Paso 2

3. Donde dice “Juego de caracteres del archivo” seleccione del menú desplegable la opción “latin1”.

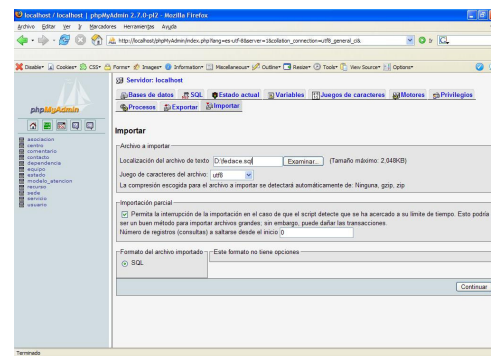


Figura D-6 Recuperar copia de seguridad. Paso 3

4. Pulse el botón “Examinar” y localice el archivo con la copia de seguridad.
5. Pulse el botón “Continuar”.

Si todo ha salido correctamente, se notificará con un mensaje de confirmación.